

UNIVERSIDADE DE LISBOA  
FACULDADE DE CIÊNCIAS  
DEPARTAMENTO DE INFORMÁTICA



## **Desenvolvimento de soluções à medida em Sharepoint**

Pedro Davim Teixeira Mendes

**Mestrado em Engenharia Informática**  
Especialização em Engenharia de Software

Trabalho de projeto orientado por:

Prof. Doutor Luís Antunes

Eng. André Alves

2018



# Agradecimentos

Após um longo percurso académico tenho o orgulho de agradecer à minha MÃE que nunca me deixou desistir de continuar a estudar, que sempre me deixou seguir o meu caminho, mas nunca me deixou seguir por um mau caminho, que sempre me soube confortar ou chamar à atenção nas alturas certas. Obrigado MÃE.

Tenho igualmente o orgulho de agradecer ao meu PAI que sempre esteve presente na minha vida para me mostrar quais os melhores caminhos a seguir e por me dar “o bichinho” da Informática que hoje me permite terminar este mestrado. Obrigado PAI.

Quero agradecer aos meus irmãos mais novos por todas as alegrias e arrelias que me deram pois fazem com que me torne melhor pessoa. Obrigado aos meus Irmãos

Quero agradecer às minhas avós e tia por sempre acreditarem no meu sucesso e quererem que chegue sempre mais longe. Obrigado às minhas avós e tia.

Quero agradecer aos amigos que criei durante esta jornada e que tão importantes foram e continuam a ser para que consiga alcançar os meus objetivos. Obrigado ao Paulo Martins, Ricardo Cardoso, Dário Ferreira, Andreia Santos, Adriana Santos, Roberto Franco, André Alves e Ivan Carneirinho.

Quero agradecer aos meus colegas de trabalho que incansavelmente me orientaram, aconselharam e ensinaram a ser um melhor profissional. Obrigado à Inês Oliveira, ao Luís Martins, ao Pedro Nunes e todos os outros colegas da Unipartner.

Quero agradecer ao Sasha por sempre me orientar e ajudar durante o meu percurso académico. Obrigado também ao Faneca, Famel, Ametista, Gaga e toda “La Famiglia”.

Quero agradecer aos meus orientadores, de estágio, por ouvirem as minhas dúvidas e incertezas e indicarem-me caminhos alternativos de modo a obter sucesso. Obrigado ao André Alves e um especial agradecimento ao Professor Doutor Luís Antunes.

Por fim quero deixar o meu maior agradecimento à pessoa com que mais intimamente partilhei os meus fracassos, mas também os meus sucessos, partilhei os melhores e piores momentos da minha vida pessoal. Obrigado à minha linda namorada, obrigado à Inês Santos.

Obrigado à Cookie e Sininho. “Au-Au!!”



## Resumo

As empresas com um elevado número de pessoas na sua estrutura deparam-se frequentemente com problemas de colaboração entre os trabalhadores. Os problemas mais comuns são a falta de organização e controlo sobre a partilha de informação e o mau manuseamento de dados e informações. Nos dias de hoje, existem diversas tecnologias para colmatar estes problemas oferecendo soluções e funcionalidades *out-of-the-box* que nem sempre coincidem com os requisitos de todas as empresas. Neste relatório de estágio são descritas duas soluções produzidas, com recurso às tecnologias *web*, para adicionar funcionalidades ao Sharepoint e desta forma satisfazer os requisitos do cliente. As funcionalidades foram adicionadas através da criação de páginas *web* e aplicação de estilos e lógica de negócio às mesmas. A primeira solução tem como objetivo permitir que os colaboradores do cliente reportem as suas horas. A segunda solução tem como objetivo administrar formação aos colaboradores através de *e-learning*. As soluções foram desenvolvidas sobre tecnologias Microsoft, em especial o Sharepoint *online* na solução de reporte de horas e o Sharepoint *online* e *on-premises* na solução de *e-learning*. A computação na nuvem permite reduzir os custos de produção de *software*, pois todos os custos de gestão e manutenção de infraestruturas são geridos por empresas especializadas. No entanto, certas empresas não estão totalmente preparadas para fazer a passagem completa das suas infraestruturas para a nuvem, mesmo quando esta tecnologia apresenta vantagens muito fortes para impulsionar os seus negócios. E estes projetos para o mesmo cliente são exemplo disso.

**Palavras-chave:** Microsoft, Sharepoint online, Sharepoint on-premises, Javascript, desenvolvimento à medida.



# Abstract

Companies with many people in their structure often have problems of collaboration between workers. The most common problems are lack of organization and control over shared information and poor handling of data and information. Today, there are several technologies to address these issues by offering out-of-the-box solutions and functionalities that do not always match the requirements of each business. In this internship report, two solutions are described, using web technologies to add functionality to Sharepoint and so satisfy the client's requirements. The features were added through the creation of web pages and application of styles and business logic to them. The first solution is to allow the client's employees to report their hours. The second solution aims to train employees through e-learning. The solutions were developed on Microsoft technologies, Sharepoint online in the time-reporting solution and online Sharepoint and on-premises in the e-learning solution. Although cloud computing can reduce software production costs, because all infrastructure management and maintenance costs are managed by specialized companies, certain companies are not fully prepared to pass their infrastructures to the cloud, even though this technology has very strong advantages to boost their business. And these projects for the same client are examples of this.

**Keywords:** Microsoft, Sharepoint online, Sharepoint on-premises, Javascript, custom development.





# Índice

Capítulo 1	Introdução.....	1
1.1	Introdução .....	1
1.2	Motivação .....	1
1.3	Objetivos.....	2
1.4	Organização do documento .....	2
Capítulo 2	Trabalho relacionado.....	5
2.1	Portais .....	5
2.1.1	Motivação para utilização de portais.....	5
2.2	Sharepoint.....	6
2.2.1	Distribuições do Sharepoint .....	7
2.2.2	SharePoint Foundation .....	7
2.2.3	SharePoint Server.....	8
2.2.4	Arquitetura dos <i>Sites</i> de SharePoint .....	11
2.3	JavaScript .....	12
2.3.1	O que é o Javascript?.....	12
2.3.2	Jquery .....	13
Capítulo 3	Solução de reporte de horas .....	15
3.1	Planeamento .....	15
3.2	Arquitetura.....	16
3.2.1	Vista de módulos.....	17
3.2.2	Modelo de dados .....	22
3.3	Comunicação entre o <i>front-end</i> e o Sharepoint .....	26
3.4	<i>Front-end</i> .....	27
3.4.1	Página de gestão de períodos .....	29
3.4.2	Página inicial.....	32
3.4.3	Páginas de listagem de informações.....	33
3.4.4	Folha de Horas .....	38

3.4.5	Página da folha de alocações internas .....	40
3.5	Backend .....	42
3.6	Entrada em produção e apoio à mudança .....	43
Capítulo 4	Solução de <i>e-learning</i> .....	45
4.1	Arquitetura .....	46
4.1.1	Estrutura da coleção de sites “Forma” .....	49
4.1.2	Estrutura da coleção de <i>sites</i> “Forma Administração” .....	51
4.2	Comunicação entre o Módulo de Integração e o <i>Front-end</i> .....	53
4.3	<i>Front-end</i> .....	53
4.3.1	Inscrição em cursos .....	54
4.3.2	Confirmar Sugestões .....	55
4.3.3	Obter Sugestões .....	57
4.3.4	Incumpridores de objetivos gerais .....	58
4.3.5	Configuração de sugestões .....	59
Capítulo 5	Conclusões .....	61
5.1	Trabalho desenvolvido .....	61
5.2	Dificuldades .....	61
5.3	Conclusões .....	61
5.4	Trabalho futuro .....	63
Capítulo 6	Bibliografia .....	65

# Lista de Figuras

Figura 1- Exemplo de um site de Sharepoint.....	6
Figura 2 - Arquitetura de um site de Sharepoint.....	11
Figura 3 - Arquitetura da solução de reporte de horas .....	17
Figura 4 - Módulos de HTML .....	18
Figura 5 - Módulos de CSS.....	18
Figura 6 - Módulos de JavaScript .....	20
Figura 7 - Modelo de dados .....	21
Figura 8- modelo de pedido CSOM.....	26
Figura 9 - exemplo da criação de pedido CSOM.....	27
Figura 10 - exemplo de execução e utilização da resposta de um pedido CSOMI .	27
Figura 11 - Peças Web do Sharepoint.....	28
Figura 12 - template de html para páginas .....	28
Figura 13 - Template de um ficheiro JavaScript.....	29
Figura 14 - Página de gestão de períodos .....	30
Figura 15 - Diagrama de atividades da página de gestão de períodos .....	31
Figura 16 - Página inicial da solução .....	32
Figura 17 - Diagrama de atividades da página inicial.....	33
Figura 18 - Página de listagem de folhas de horas.....	35
Figura 19 - Página de listagem de tarefas .....	36
Figura 20 - Página de relatórios .....	37
Figura 21 - Diagrama de atividades da página de listagem de folhas de horas .....	38
Figura 22 - Página das folhas de horas .....	39
Figura 23 - Diagrama de atividades da página das folhas de horas .....	40
Figura 24 - Página da folha de alocação interna .....	41
Figura 25 - Diagrama de atividades da folha de alocação interna .....	42
Figura 26 - Componentes da solução de e-learning .....	46
Figura 27 - Estrutura da coleção de sites "Forma" .....	49
Figura 28 - Exemplo da peça web "Dashboard" .....	50

Figura 29 - Catálogo de cursos .....	50
Figura 30 - Estrutura da coleção de sites "Forma Administração" .....	51
Figura 31 - Vista de módulos para a coleção de site "Forma Administração" .....	52
Figura 32 - Função responsável por criar pedido Ajax.....	53
Figura 33 - página de inscrição em cursos .....	54
Figura 34 - Diagrama de atividades da página inscrição em cursos .....	55
Figura 35 - Página de confirmação de sugestões .....	56
Figura 36 - Diagrama de atividades da página "Confirmar Sugestões" .....	57
Figura 37 - Página para obter sugestões de cursos .....	58
Figura 38 - Página para listar incumpridores de objetivos.....	59
Figura 39 - Página de configuração de Sugestões.....	60

# Lista de Tabelas

Tabela 1 - Funcionalidades do módulo de integração.....	48
---	----



# **Capítulo 1**

## **Introdução**

### **1.1 Introdução**

O presente relatório de estágio é elaborado no âmbito da disciplina Projeto de Engenharia Informática, da Faculdade de Ciências da Universidade de Lisboa, com vista à conclusão do Mestrado em Engenharia Informática.

O estágio foi feito na Unipartner,S.A., com sede em Oeiras, Lisboa. O mesmo teve a duração de nove meses, com início a 31 de outubro de 2016 e término a 31 de julho de 2017.

O estágio foi uma excelente oportunidade para desenvolver as minhas qualidades socioprofissionais e aplicar, na prática, os conhecimentos que adquiri ao longo do meu percurso académico na Faculdade de Ciências.

Este relatório contém uma descrição e análise crítica daquilo que foi feito por mim ao longo do estágio.

### **1.2 Motivação**

O Sharepoint é uma ferramenta com muitas funcionalidades desenvolvidas e que satisfaz os requisitos de colaboração, que são bem conhecidos das empresas. Contudo, não há empresas iguais, mesmo que sejam da mesma área de negócio. Duas empresas podem ter requisitos semelhantes e os pontos de variação não serem cobertos pelas funcionalidades do Sharepoint.

O cliente é uma organização que, para além dos requisitos bem conhecidos das empresas, também tem requisitos muito específicos de negócio. Esta situação cria uma necessidade de desenvolver funcionalidades adicionais para complementar a ferramenta. O estágio surgiu da necessidade da elaboração de dois projetos nos quais se pretende utilizar ao máximo as funcionalidades do Sharepoint e quando não for possível utilizar para os requisitos mais específicos, desenvolver as funcionalidades que o expandem.

## 1.3 Objetivos

Os projetos em que estive envolvido fazem parte de um alargado plano de negócio que, pretende reduzir custos e aumentar a produtividade dos colaboradores envolvidos nos diferentes processos. O projeto de reporte de horas teve como objetivo a distribuição de custos pelos diversos departamentos do cliente, sendo esta alocação de custos feita por todos os colaboradores. O projeto de *e-learning* teve como objetivo administrar formações aos colaboradores, captar os resultados destas formações para numa fase posterior sugerir cursos e atribuir prémios de produtividade.

Em termos tecnológicos, os projetos tentam os seguintes objetivos:

- Maximizar a utilização das funcionalidades e minimizar as personalizações do Sharepoint para produção das soluções;
- Criar interfaces responsivas através da utilização de tecnologias para desenvolvimento na *web*, tais como: Javascript, CSS e Html;
- Aumentar a produtividade dos colaboradores, reduzindo ao máximo o tempo e número de passos necessários para a execução de procedimentos;
- Disponibilizar as soluções na nuvem, para que os colaboradores possam escolher os momentos e os meios mais adequados a realizar os procedimentos.

## 1.4 Organização do documento

O relatório final é composto da seguinte forma:

- Capítulo 2 Trabalho relacionado – Este capítulo começa por introduzir as tecnologias de portais e justifica as vantagens da sua utilização. Apresenta a ferramenta de portais da Microsoft e descreve as suas diferentes vertentes, bem como as funcionalidades que disponibiliza. Por fim faz uma abordagem histórica do JavaScript seguida de uma definição da tecnologia e da apresentação de uma biblioteca que foi intensamente utilizada na construção das soluções;
- Capítulo 3 Solução de reporte de horas – Neste capítulo é feita uma descrição detalhada do primeiro projeto realizado, no âmbito do estágio. A descrição é feita da seguinte forma:
  1. Explicação do contexto do projeto e das regras de negócio;
  2. Descrição do plano de projeto;
  3. Descrição da arquitetura da solução com recurso a diferentes diagramas;



4. Explicação dos métodos de implementação das soluções;
  5. Descrição do trabalho realizado após a entrada em produção da solução.
- Capítulo 4 Solução de *e-learning* – À semelhança do capítulo anterior, a descrição desta solução é feita da seguinte forma:
    1. Explicação do contexto do projeto e das regras de negócio;
    2. Descrição da arquitetura da solução com recurso a diferentes diagramas;
    3. Explicação dos métodos de implementação das soluções;
    4. Descrição do trabalho realizado após a entrada em produção da solução.
  - Capítulo 5 Conclusões – Neste capítulo concluo o relatório de estágio fazendo uma análise crítica ao estágio e ao trabalho por mim desenvolvido e faço uma descrição de possíveis desenvolvimentos futuros.



## Capítulo 2 Trabalho relacionado

### 2.1 Portais

Um portal empresarial é uma forma de as empresas disponibilizarem rápidos acessos a informação por parte dos seus colaboradores. Em muitos casos, um portal consiste em vários *websites*, sendo que a informação fica armazenada diretamente nesses *sites* ou em outros sistemas. Estas tecnologias permitem que os colaboradores tomem decisões informadas e por isso torna-se necessário que a informação usada esteja protegida, atualizada e facilmente acessível.

#### 2.1.1 Motivação para utilização de portais

1. Hoje em dia, os colaboradores têm um grande acesso às novas tecnologias (computadores, tablets e smartphones), por isso a expectativa de que o seu trabalho será realizado nas mesmas tecnologias é elevada;
2. Sendo os portais uma tecnologia baseada na internet, toda a informação pode estar disponível independentemente da localização do colaborador;
3. Permite que um colaborador execute as suas tarefas do início ao fim, no mesmo local;
4. Permite aos auditores realizarem auditorias com bastante precisão porque são registadas operações realizadas sobre os dados;
5. A forma de partilhar informação passa a ser definida pela empresa e não pela pessoa que a partilha, assim os colaboradores sabem sempre como esta pode ser encontrada;
6. As tecnologias de portais são escaláveis e por isso estão sempre prontas para acompanhar o crescimento das empresas;
7. As tecnologias de portais permitem a integração com sistemas obsoletos mas que são necessários ao funcionamento das empresas [1].

## 2.2 Sharepoint

O SharePoint é uma tecnologia de portais. Permite às empresas de todas as dimensões aumentar a produtividade das equipas e aumentar a eficiência dos processos de negócio. Para isso, fornece um grande conjunto de ferramentas para organizar conteúdos, gerir documentos, partilhar conhecimento, oferecer ambientes de colaboração robustos e encontrar informações e pessoas. Algumas ferramentas são disponibilizadas em *webparts*, que são blocos de conteúdos que são adicionados a uma página de Sharepoint. Na Figura 1, podemos visualizar um exemplo de um *site* de Sharepoint com duas colunas. Na primeira coluna, foram adicionadas duas *webparts* diferentes: os *announcements*, que permitem adicionar avisos, e os *useful links*, que apontam para uma lista com endereços importantes para os utilizadores.

HOMEDEPARTMENTS▼DOCUMENTS▼PROJECTSWIKI▼EXTERNALSEARCHEDIT LINKS


ANNOUNCEMENTS

Title

Register for Christmas Party!


...


Body



Benefits Open Enrollment

...





COMPANY EVENTS

June 2017

SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
28	29 10:00 pm SharePoint	30 10:00 pm Company	31	1 12:00 pm - Lunch & L	2	3
4	5 10:00 pm SharePoint	6	7	8 12:00 pm - Lunch & L	9	10 + Add
11	12 10:00 pm SharePoint	13 10:00 pm Company	14	15 12:00 pm - Lunch & L	16	17
18	19 10:00 pm SharePoint	20	21	22 12:00 pm - Lunch & L	23	24
25	26 10:00 pm SharePoint	27 10:00 pm Company	28	29 12:00 pm - Lunch & L	30	1

OUR BLOG

Blog – SharePoint Maven

How to set up retention and deletion policies for files and folders in SharePoint

Why your SharePoint Implementation Failed

Solve all your SharePoint sync issues by using the new OneDrive Sync Client

USEFUL LINKS

+ new link or edit this list

✓ URL

Company Website ...

Figura 1- Exemplo de um site de Sharepoint

### 2.2.1 Distribuições do Sharepoint

O Sharepoint é distribuído na forma *on-premises* ou na nuvem.

Na primeira forma, é responsabilidade da empresa construir, configurar e gerir todo o ambiente Sharepoint. A empresa pode fazê-lo em servidores internos dedicados para o efeito ou então pode fazer a mesma instalação nos servidores de um parceiro, sendo este o responsável pela manutenção da infraestrutura.

Na segunda forma, a empresa necessita de realizar uma subscrição do *Microsoft Office 365*. Os serviços são hospedados e geridos pela *Microsoft*, nos seus *data centers*. Desta forma, a empresa não tem a preocupação de manter as infraestruturas nem de fazer instalações, configurações ou atualizações do sistema.

### 2.2.2 SharePoint Foundation

O Sharepoint Foundation é uma das versões do Sharepoint *on-premises* e oferece as funcionalidades fundamentais de uma ferramenta de colaboração que são descritas de seguida.

Relativamente à colaboração, este disponibiliza:

- *Sites* de equipa - para partilha de documentos, coordenação de calendários, coordenação de tarefas e participação em fóruns de discussão;
- Bibliotecas de documentos – um ambiente para a gestão e criação de documentos, bem como o controlo de versões destes;
- Permissões – os acessos aos documentos são verificados ao nível do utilizador e de acordo com as funções que este tem;
- Sistemas de alerta – permitem manter os colaboradores atualizados das suas tarefas correntes e de novas tarefas;
- *Templates* – permitem criar estruturas de conteúdos para posterior utilização e com isso acelerar a criação e a partilha.

É possível diminuir custos de implementação e de recursos porque:

- A criação de *sites* de equipa é extremamente fácil e desta forma não é necessário ter recursos humanos especializados para realizar esta tarefa;
- É possível aplicar diferentes estilos personalizados aos *sites* sem sair do *browser*;

- A edição dos conteúdos é feita em ferramentas do Microsoft Office (Word, Excel, PowerPoint), visto que a integração entre estes é perfeitamente consistente.

A organização e controlo dos dados sensíveis da empresa são mais facilitados porque o SharePoint Foundation oferece:

- Comandos de aprovisionamento de *sites*, gestão de conteúdos, suporte e *backup* de dados. Estes comandos estão disponíveis tanto numa linha de comandos como no *browser*;
- Funcionalidades administrativas que permitem definir quais são as permissões das diferentes equipas, tanto a nível de acessos como a nível de criação de *sites*;
- A recuperação de dados apagados ou de versões anteriores dos mesmos. Estes dados antes de serem apagados definitivamente ficam por um período de tempo numa lista chamada “Recycle Bin”.

### 2.2.3 SharePoint Server

O SharePoint Server expande o SharePoint Foundation através de um conjunto de funcionalidades mais avançadas. As principais funcionalidades que este oferece são explicadas de seguida.

A pesquisa no SharePoint facilita o trabalho dos colaboradores em encontrar informação que está espalhada pelos diferentes sites e *sub-sites* de SharePoint ou em outras fontes de informação. Os componentes que constituem a pesquisa do SharePoint são os seguintes:

- Origem dos conteúdos – de forma a que o SharePoint consiga procurar pela informação, este deve ter configurado uma origem de conteúdos que serão procurados. As origens de conteúdos podem ser: *Sites* de Sharepoint, *sites* que não do SharePoint, repositórios de partilha de ficheiros, pastas públicas do Microsoft Exchange e aplicações empresariais externas a este;
- Índices – os detalhes de todos os conteúdos adicionados à origem de conteúdos são colocados no índice. De forma a que a pesquisa dos conteúdos seja precisa e gere resultados, o sistema deve com frequência atualizar o índice;
- *Crawling Schedule* – é um processo que o SharePoint usa para percorrer os conteúdos e construir o índice de pesquisa. Podem ser feitos dois tipos de

*crawl*: de forma incremental, sempre que houver uma mudança ou um ficheiro novo ou de forma completa, geralmente usado quando há mudanças significativas de configuração do sistema de pesquisa;

- Pedidos - os pedidos são executados sobre o índice para identificar os conteúdos que contém ligações com as palavras ou frases pesquisadas. O resultado desse pedido é uma lista de resultados;
- Origem de resultados – esta componente guarda a informação de como o SharePoint deve selecionar a informação que a *query* pode encontrar no índice;
- Tipos de resultados – permite que os administradores possam customizar a forma como os documentos são apresentados aos utilizadores de acordo com o tipo de documento.

A gestão de conteúdos *web* é um processo para criar e gerir conteúdos tanto na Internet como na Intranet. Para suportar uma plataforma de gestão de conteúdos *web*, o Sharepoint oferece as seguintes funcionalidades:

- Estilos e páginas mestras – desta forma é possível manter a consistência entre todos os *sites* e de cada vez que uma página for criada, será igual;
- Criação de páginas no *browser* – permite que utilizadores sem conhecimento de tecnologias para criação de páginas consigam criar páginas com o aspeto do *site*;
- Fluxos de trabalho - processo automático para aprovação e publicação de conteúdo;
- Versionamento dos conteúdos - garante que os colaboradores trabalham sempre na versão mais recente do documento e oferece a possibilidade de restaurar versões anteriores, uma vez que preserva um registo das alterações;
- Relatórios – oferece uma visão geral dos conteúdos;
- *Check-In e Check-out* – Mecanismo para controlar e assegurar que várias pessoas a trabalhar em simultâneo num documento estão sempre a trabalhar no mesmo documento.

Os serviços *enterprise* são formas de construir soluções utilizando ferramentas do Microsoft Office. A utilização destas ferramentas permite acelerar o processo de criação

das soluções bem como facilitar essa criação. Os serviços que podem ser usados para isso são:

- Serviços *InfoPath Form* – é uma aplicação que integra com Sharepoint, cuja função é recolha e apresentação de dados. Estes formulários são documentos de XML e podem ser desenhados para recolher informação de forma a complementar as formas de recolha de dados do Sharepoint.
- Serviços de Excel, Visio e PerformancePoint – O serviço de Excel permite apresentar os dados de um ficheiro Excel diretamente numa *webpart*. O serviço Visio disponibiliza diagramas dentro de *webparts*, sendo que os dados podem ser recolhidos de diferentes origens. O serviço de PerformancePoint permite criar painéis com informação sobre as diferentes aplicações das empresas.
- Serviços Access – Este serviço possibilita a interação das *webparts* com o Microsoft Access, que é um produto de bases-de-dados relacionais.[1]

*Business Connectivity Services (BCS)* constituem uma plataforma centralizada do Sharepoint que permite a criação de soluções que necessitem de integração de dados com outras aplicações. Estes serviços podem ser usados como interface de apresentação dos dados que são obtidos através de *web services*, fontes *OData*, entre outros.[2]

*Records Management* define-se por manter e organizar conteúdos de acordo com determinadas regras. Estas regras definem onde os ficheiros serão guardados, durante quanto tempo devem ser mantidos, como devem ser eliminados e quem é responsável pelos mesmos. Um sistema de *Records Management* é composto pelos seguintes elementos:

- Análise de conteúdos - descreve e categoriza os conteúdos que poderão fornecer evidências;
- Plano de ficheiros – define para cada tipo de ficheiros quando estes devem ser retidos como evidências, quanto tempo devem ser retidos, como devem ser descartados e quem é responsável por mantê-los;
- Documento de requisitos de conformidade – define as regras que os sistemas devem seguir para assegurar a conformidade bem como as metodologias a serem usadas;
- Métodos para recolher evidências que não estão ativas, para fazer auditorias e para recolher meta-dados das mesmas;



- Processo de retenção de evidências para a eventualidade de acontecerem litígios;
- Um sistema para reporte e manutenção de evidências, para garantir que os colaboradores preencham, acedem e gerem as mesmas como definido pelas políticas e processos que estão definidos [3].

#### 2.2.4 Arquitetura dos *Sites* de SharePoint

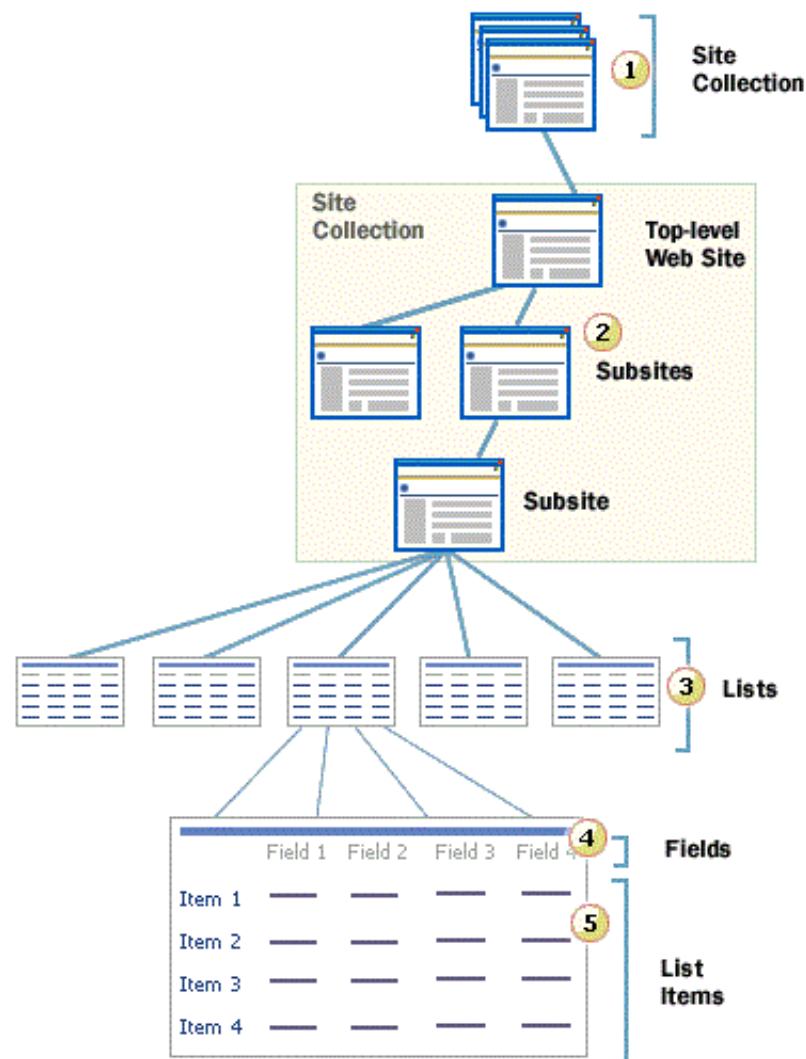


Figura 2 - Arquitetura de um site de Sharepoint

Na Figura 2, está representada a arquitetura hierárquica de um ou mais *sites* de Sharepoint. No topo, está a coleção de *sites* que permite isolar configurações entre duas coleções diferentes. As configurações que estão isoladas são: definição de metadados; navegação das páginas; grupos de permissões; *templates* e estilos das páginas [4].

Dentro de uma coleção há um *site* principal que pode conter páginas ou *subsites*. Cada *subsite* pode conter, tal como o anterior, um conjunto de páginas e *subsites*. Desta forma é possível montar uma estrutura no Sharepoint que reflita a estrutura da organização que necessitar de uma ferramenta de colaboração. Dentro de cada *site* da estrutura é possível criar listas. As listas têm como objetivo guardar informação que podem ser documentos, ficheiros, endereços ou valores (texto, números, datas, etc). Em cada lista é possível adicionar colunas para refletir as diferentes propriedades dos dados. Por fim, quando um item é criado na lista, as colunas podem ou não ser preenchidas com valores.

## 2.3 JavaScript

O JavaScript foi criado em maio de 1995 por Brendan Eich. O seu primeiro nome foi Mocha. A setembro do mesmo ano o nome foi alterado para LiveScript, mas a verdadeira ação de *marketing* aconteceu quando a linguagem adotou o nome de JavaScript em dezembro de 1995. Na altura da última mudança, a linguagem de programação Java, estava a tornar-se muito popular.[5]

Em Novembro de 1996, foi iniciado o desenvolvimento do standard ECMAScript pela European Computer Manufacturers Association (ECMA) com base em várias tecnologias, sendo as mais conhecidas o JavaScript, que nesta altura era utilizado pela Netscape, e o JScript, que era utilizado pela Microsoft.[6]

O desenvolvimento do ECMAScript foi feito em ciclos e em 1998 foi lançado o ECMAScript 2 e no ano seguinte foi lançada uma terceira versão. A terceira versão adicionou ao standard expressões regulares melhores, melhor gestão de strings, novas expressões de controle, gestão de exceções através de “try/catch”, melhores definições dos erros, entre outros. Estas alterações na terceira versão foram decisivas para que todos os *browsers* suportem a utilização do Javascript.[5], [6]

Atualmente, o ECMAScript está na sua oitava edição e o Javascript na versão 1.8.5, sendo que esta versão apenas está em conformidade com a quinta versão do ECMAScript [7], [8].

### 2.3.1 O que é o Javascript?

O Javascript é uma linguagem de programação dinâmica, orientada a objetos, imperativa e declarativa. É baseada em protótipos, ou seja, as classes não são definidas explicitamente, são construídas através da adição de métodos e propriedades a

instâncias de outros objetos. Apesar de ter sido originalmente desenhada para ser utilizada em *browsers web*, onde hoje em dia é massivamente usada, esta também pode ser utilizada em ambientes que não o *browser*. O Javascript pode ser uma linguagem interpretada pelo ambiente que a executa, sendo que neste caso não há a necessidade de ser compilada, ou pode ser compilada durante a execução do programa e neste caso as funções são tratadas como variáveis que podem ser passadas por parâmetro a outras funções, retornadas dentro de uma função ou ser alteradas durante a execução [9,10,11].

### 2.3.2 JQuery

É uma biblioteca de Javascript que adiciona inúmeras funcionalidades orientadas à manipulação de HTML, gestão de eventos, animações e execução e tratamento de pedidos Ajax (*Asynchronous JavaScript and XML*). Usada para auxiliar o desenvolvimento de aplicações *web*, esta biblioteca permite acelerar o desenvolvimento pois reduz o número de linhas de código [12].

As funcionalidades desta biblioteca apenas estão disponíveis quando o ficheiro do *script* é carregado na página. A utilização do JQuery é feita através da invocação de uma função chamada “jquery” ou por uma abreviatura “\$”. Para assegurar uma correta utilização desta biblioteca apenas deve ser usada quando o documento de HTML estiver pronto para ser manipulado e é através da função “\$(document).ready()” que conseguimos identificar esse momento. Na função “ready()” podemos passar uma função por parâmetro que será executada quando o documento estiver pronto a ser manipulado [12,13].

Para manipular o ficheiro DOM (*Domain Object Model*) de uma página HTML através da biblioteca de JQuery são utilizados seletores. Estes seletores podem ser combinações de tipos de *tags* (div, h1, h2, etc...), identificadores de *tags* e classes de estilos das *tags*. Adicionalmente, é possível organizar seletores que selecionem vários elementos do DOM que estão em diferentes ramos e diferentes níveis nos ramos[14].

Na gestão de eventos, a biblioteca oferece o método “on()” que permite associar eventos a objetos jquery. O primeiro parâmetro é uma *string* que define o evento que se quer associar (*click*, *submit*, *change*, etc...). O segundo parâmetro poderá ser um objeto JSON (JavaScript Object Notation), cujo objetivo é passar informação para a função que é executada quando o evento acontece ou se não houver a necessidade de passar informação para a função será a própria função [15].

A realização de pedidos AJAX é feita através da função “ajax()”. Esta função permite a criação do pedido de HTML e definição de funções a serem chamadas para as possíveis diferentes respostas dos pedidos. A criação do pedido é feito através da construção de um objeto JSON que é passado por parâmetro para a função “ajax()” [16].

Aliadas às funcionalidades descritas anteriormente, esta biblioteca é massivamente utilizada no desenvolvimento de aplicações *web* porque apresenta as seguintes vantagens: funções extremamente otimizadas e testadas, suporte da comunidade e menos preocupações para os programadores que necessitam de desenvolver para os diferentes *browsers*.

## Capítulo 3 Solução de reporte de horas

Este projeto surge da necessidade do cliente em perceber onde e como é que os seus colaboradores estão a utilizar o seu tempo, sendo o objetivo final imputar custos de serviços prestados de forma correta.

O reporte das horas é feito diariamente em períodos semanais, ou seja, todas as semanas os colaboradores deverão aceder ao *web site* da solução e introduzir os seus dados.

O reporte de horas é feito através da identificação do esforço que tiveram para realizar as suas tarefas. Se em um dia o colaborador realizou duas tarefas, este deve dividir o esforço por essas tarefas, em percentagem.

### 3.1 Planeamento

Este projeto durou aproximadamente quatro meses e meio. Durante este tempo foram realizadas seguintes tarefas:

- Análise;
- Implementação;
- Testes;
- Criação de material de apoio e formações;
- Suporte após entrada em produção.

A implementação da solução dividiu-se em dois ciclos. O ciclo de implementação neste projeto caracteriza-se por:

1. Desenvolvimento de um conjunto de funcionalidades em ambiente de desenvolvimento;
2. Criação de conteúdos de apoio ao utilizador;
3. Passagem das funcionalidades para o ambiente de qualidade;
4. Passagem de conhecimento, ao cliente, sobre o funcionamento das funcionalidades;

## 5. Testes de aceitação.

Foram realizadas formações presenciais para todos os utilizadores finais da solução, 8 foram realizadas em Lisboa e 3 no Porto. Estas formações têm o objetivo de transmitir a importância de realizar o controlo do tempo gasto e de como é que o podem realizar na solução desenvolvida. Aliadas a estas formações foram disponibilizados aos utilizadores:

- Um manual com instruções sobre o funcionamento da solução bem como esclarecimentos sobre a sua necessidade;
- Cartazes de anúncio com informações sobre a data de início de utilização, forma de acesso e contacto disponíveis para esclarecimento de dúvidas.

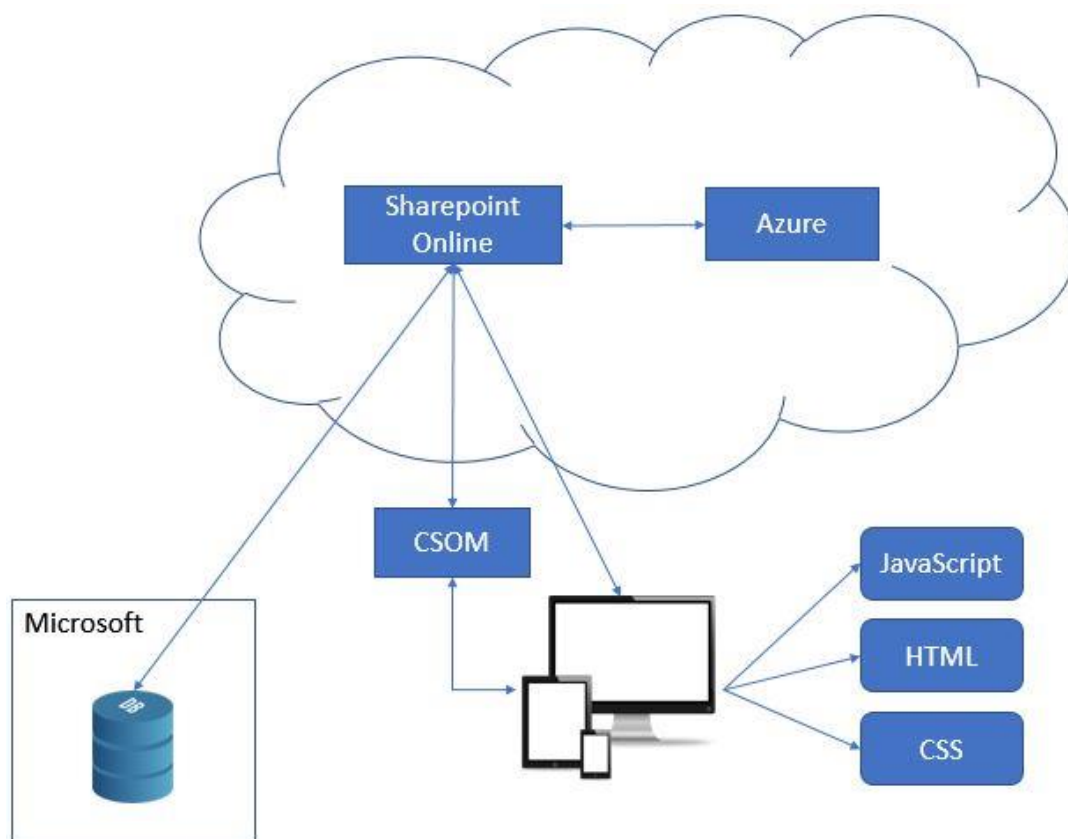
Adicionalmente, foram enviados 4 e-mails no período inicial de utilização da solução com conteúdo variado para reforçar as informações transmitidas durante as formações presenciais.

## 3.2 Arquitetura

A solução de reporte de horas utiliza intensivamente os serviços na nuvem oferecidos pela Microsoft, nomeadamente o Sharepoint e o Azure.

No primeiro, é montada toda a estrutura e navegação do *site* bem como o armazenamento de dados criados pelo utilizador. A gestão, obtenção e atualização de dados é feita através do *Client Side Object Model* (CSOM). Esta forma de realizar operações sobre dados no Sharepoint oferece duas grandes vantagens que permitem reduzir o tráfego na rede.

No segundo, existem processos que são executados periodicamente para realizarem tarefas inerentes ao processo de reporte de horas, por exemplo, notificações aos utilizadores ou abertura e fecho de períodos. Para executar uma tarefa é adicionado um item a uma lista de SharePoint com uma *flag* “pendente”. A cada 15 minutos, o processo varre esta lista e se encontrar algum item com a *flag* nesse estado, executa a tarefa que está definida no mesmo.



*Figura 3 - Arquitetura da solução de reporte de horas*

Devido aos requisitos do cliente, esta solução exigiu desenvolvimentos de funcionalidades personalizadas. Estas funcionalidades foram desenvolvidas recorrendo ao html para definição da estrutura estática da página, ao CSS para definir o aspeto e ao JavaScript para fazer a manipulação dos objetos DOM e para construir e executar os pedidos CSOM.

### **3.2.1 Vista de módulos**

Nesta secção são apresentados os diferentes módulos que constituem os ficheiros da solução e são explicadas as suas funções. Os módulos estão categorizados por tipo de ficheiro (html, css e JavaScript).

O html está dividido em 3 categorias que são Administrador, Managers e Colaboradores. Esta separação tem o objetivo de realçar as permissões que os diferentes Stakeholders da aplicação têm. Os Colaboradores têm acesso à página inicial, a uma página de listagem de folhas de horas e a uma página onde fazem o reporte das horas, semanalmente. Os *managers* e diretores, para além de terem os mesmos acessos que os colaboradores, também acedem a uma página de listagem de tarefas, uma página de registos diários e uma página de alocação interna. O administrador tem acesso a toda a

área de administração, onde apenas uma página é personalizada, sendo que todas as outras páginas dessa secção utilizam funcionalidades do SharePoint. A página personalizada serve para realizar a gestão dos períodos anuais e mensais.

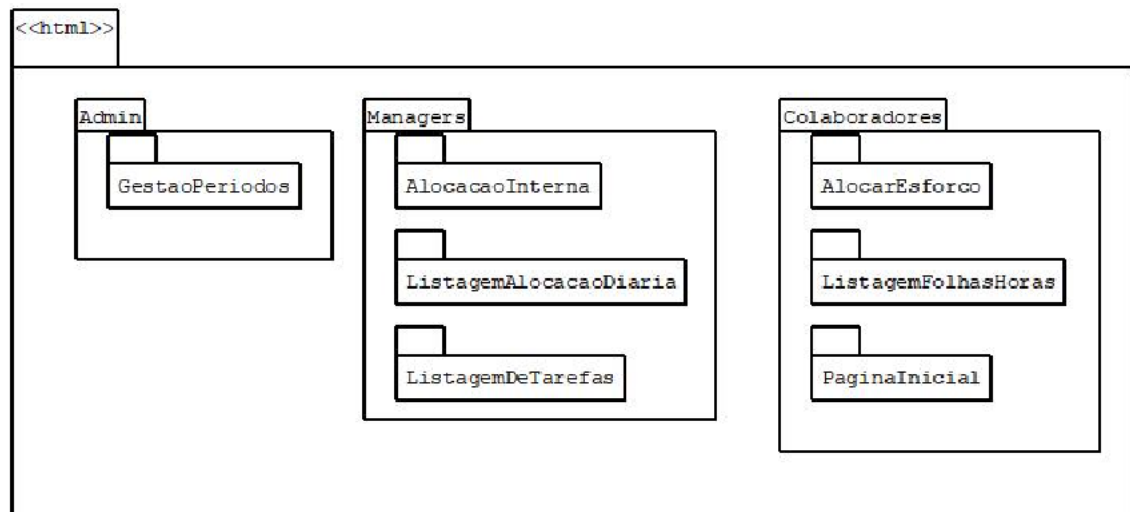


Figura 4 - Módulos de HTML

Os ficheiros CSS criados por nós para aplicar cores, fontes de texto, espaçamentos, bordas e entre outras propriedades às páginas foram o “Custom” e o “Themed”. As personalizações necessárias para transformar os estilos das bibliotecas utilizadas foram feitas no primeiro ficheiro. Por outro lado, todas as novas personalizações foram realizadas no segundo ficheiro. Os ficheiros “Chosen”, “Fabric”, “Modal” e “jQuery”, são bibliotecas externas que foram utilizadas para acelerar o desenvolvimento da solução.

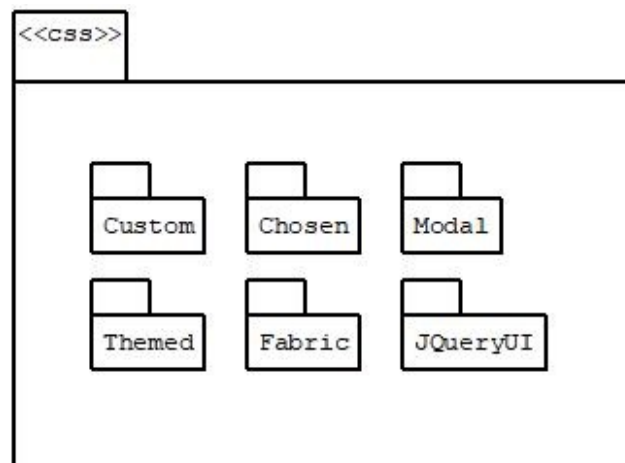


Figura 5 - Módulos de CSS



Os ficheiros de JavaScript foram organizados em 4 módulos principais, sendo eles:

- Bibliotecas - reúnem as bibliotecas de Sharepoint que oferecem funcionalidades de comunicação e interação com os dados. Também contêm a biblioteca do JQuery e bibliotecas que estendem as suas funcionalidades;
- *Data Access* - os ficheiros deste módulo estão divididos por lista ou grupos de listas, sendo que em cada ficheiro são criados e executados os pedidos CSOM. Estes pedidos são criados com recurso a um pedido CAML (*Collaborative Application Markup Language*);
- *User Interface* - este módulo agrupa os ficheiros que são responsáveis pela lógica, interação com o HTML e CSS das diferentes páginas. Cada ficheiro neste módulo representa uma página e ao todo, foram criadas 7. Em cada um destes ficheiros é feita a importação de outros scripts a serem usados, inicialização da interface gráfica, subscrição de eventos e atribuição de funções quando estes ocorrerem e ainda a inicialização de pedidos ao SharePoint;
- Utilidades - os ficheiros deste módulo contêm código que é utilizado por diferentes *scripts* da solução, de modo a poderem ser reutilizados e evitar a repetição de código no módulo anterior. O “Common” tem como principais funções mapear os endereços de listas e scripts armazenados no SharePoint. O “Utils”, disponibiliza funcionalidades genéricas como por exemplo, descarregar um ficheiro Excel ou remover itens duplicados de um *array*. O “Structures” tem como única função a criação de objetos que reflitam os campos de um item de uma lista no SharePoint. Quando uma função deste *script* é chamada, é devolvido um objeto com todas as propriedades a “null”. Se uma das propriedades for uma referência para um item de outra lista, o objeto que mapeia o item da segunda lista é incorporado no objeto inicial. O script “Pesquisa” contém uma parte da lógica das caixas de seleção presentes nas páginas de pesquisa. Neste ficheiro, são filtrados os valores que podem ser apresentados ao utilizador com sessão iniciada, definidos comportamentos de caixas com valores que dependem de outras caixas, e realizada a validação e execução da pesquisa. O *script* “DropDowns” tem uma função principal, receber um *array* de objetos, sendo que cada objeto corresponde a uma caixa de seleção da página, e popular essas caixas com os valores que recebe numa das propriedades do objeto.

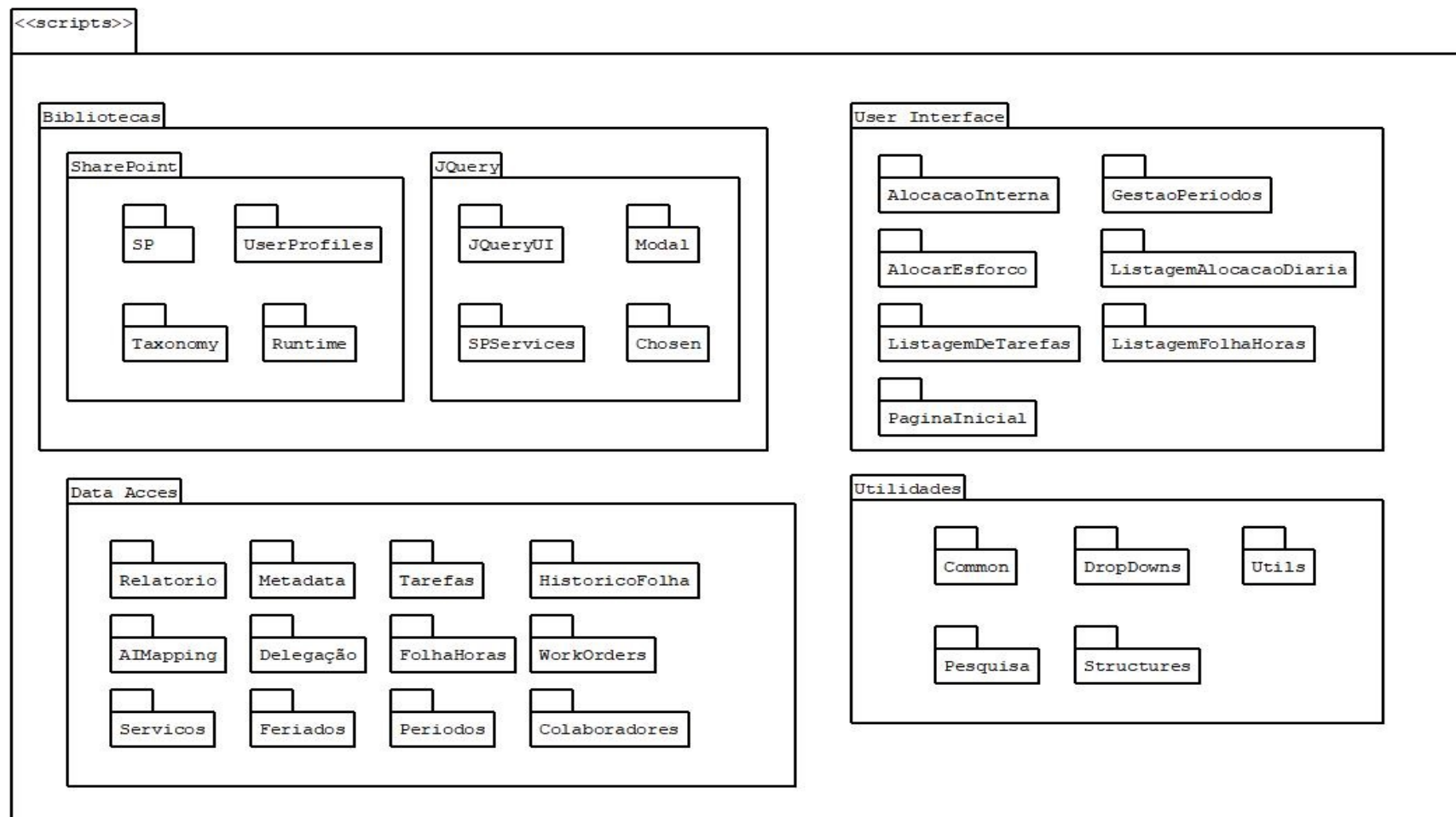


Figura 6 - Módulos de JavaScript

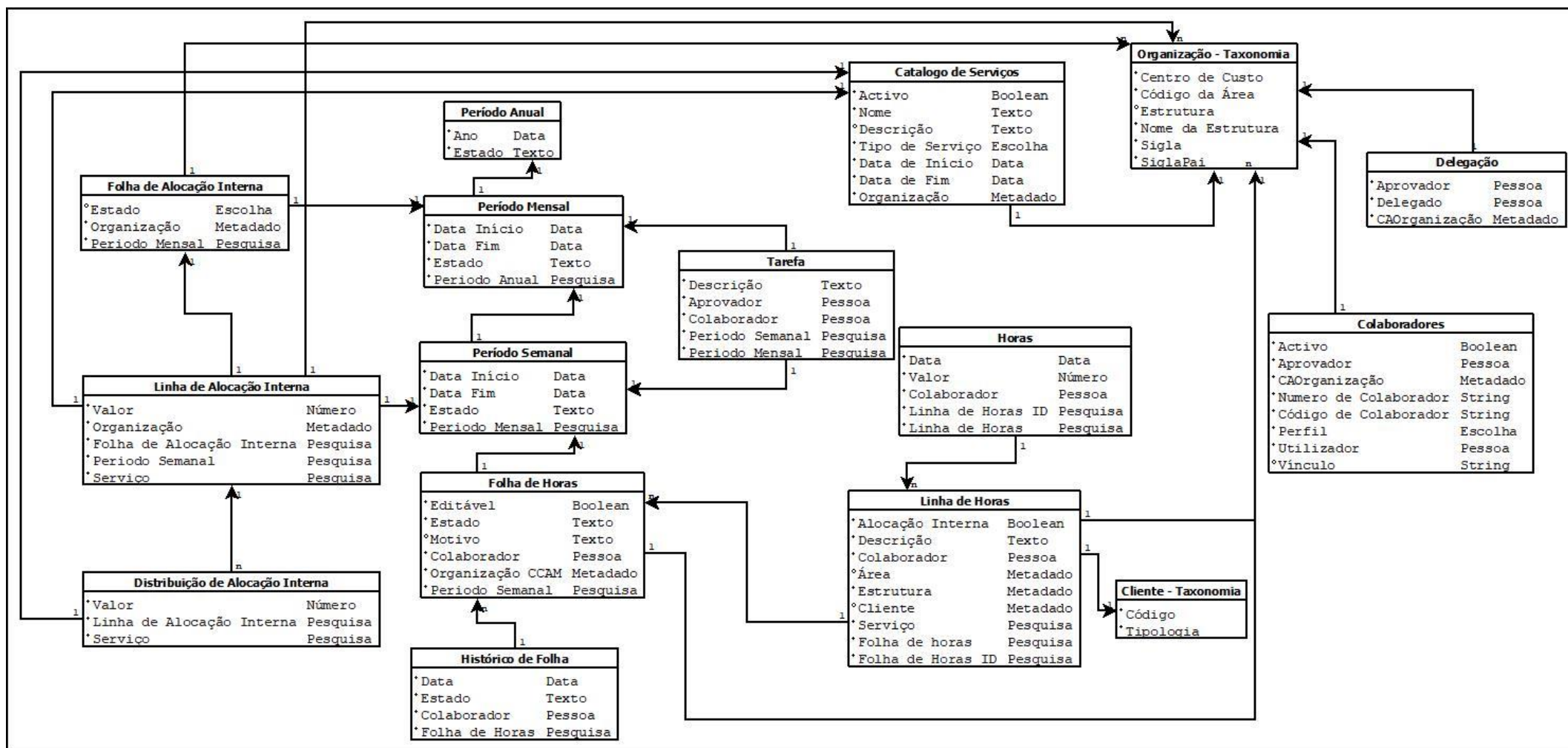


Figura 7 - Modelo de dados

### 3.2.2 Modelo de dados

A base de dados da solução foi construída com recurso a listas de Sharepoint, que permitem a adição de colunas que serão preenchidas quando um item é criado. A estas listas podem ser adicionadas colunas do tipo texto, número, data, escolha, pesquisa, booleano, pessoas/grupos e metadados[17]. No modelo de dados, Figura 7, os nomes das colunas estão listados do lado esquerdo e o tipo do lado direito. Adicionalmente a esta base de dados, foram criadas duas taxonomias que organizam conceitos de forma hierárquica. Pelo facto de ter sido usado o Sharepoint online para desenvolver e hospedar a solução, não foram necessárias preocupações, por parte da equipa de desenvolvimento, com a gestão das infraestruturas da base de dados.

A figura anterior mostra como as colunas foram criadas e como estão relacionadas entre si. A descrição das listas é a seguinte:

- Período Anual - lista que guarda os **anos** que foram abertos na aplicação e o **estado** que este tem: “aberto” ou “fechado”;
- Período Mensal – lista que guarda os meses do ano que tem o estado a “aberto”.

Os itens desta lista contêm:

- Uma **data de início** e uma **data de fim** para o período mensal;
- Um **estado** que pode ser: “Por abrir”, “Aberto”, “Em Validação” e “Fechado”;
- Uma referência à lista **período anual** e desta forma é possível saber a que ano pertence o período mensal;
- Período Semanal – lista que guarda os períodos semanais de cada mês que está na lista “período mensal”. Os itens desta lista contêm:
  - Uma **data de início** e uma **data de fim** para o período semanal;
  - Um **estado** que pode ser: “Por abrir”, “Aberto” e “Fechado”;
  - Uma referência à lista **período mensal**;
- Folha de horas – lista que cria o conceito de uma folha em que os colaboradores reportam horas. Os itens desta lista contêm:
  - Um **colaborador** ao qual a folha pertence;
  - Uma **organização** à qual o colaborador pertence;
  - Um **estado** que pode ser: “Por iniciar”, “Pendente”, “Submetida”, “Aprovada” e “Em falta”;
  - Um valor booleano que diz se a folha é **editável** ou não;

- Um **motivo** que guarda o estado anterior da folha;
- Uma referência à lista **período semanal**, pois o reporte de horas é feito semanalmente.
- Linha de Horas – lista que guarda todas as linhas de horas introduzidas nas diferentes folhas de horas. Os itens desta lista contêm:
  - Um **colaborador** ao qual a linha pertence;
  - Uma **estrutura** à qual o colaborador pertence;
  - Uma **área**, se aplicável, da estrutura do colaborador;
  - Uma **cliente**, se aplicável, para quem o colaborador prestou o serviço;
  - Uma referência à lista “Catalogo de Serviços” do **serviço** executado pelo colaborador;
  - Duas referências uma ao título da **folha de horas** e a outra ao seu **id**;
  - Uma **descrição**, se aplicável. Este campo é livre para os colaboradores escreverem notas sobre a linha adicionada;
  - Um valor booleano que indica se as horas estão a ser **alocadas internamente** ou a clientes.
- Horas – lista que guarda os valores introduzidos pelos utilizadores nas linhas de horas. Os itens desta lista contêm:
  - Um **colaborador** ao qual o registo pertence;
  - Uma **Data** em que o registo foi feito;
  - Um **Valor** que foi registado pelo colaborador;
  - Duas referências uma ao título da **folha de horas** e a outra ao seu **id**.
- Histórico de Alocação de Esforço – lista que guarda todas as ações realizadas sobre uma folha de horas. Os itens desta lista contêm:
  - Uma **Data** em que a ação foi realizada;
  - Um **Estado** que pode ser: “Submetido”, “Gravado”, “Aprovado”, “Pendente” e “Em falta”;
  - Um **colaborador** responsável pela ação realizada sobre a folha;
  - Uma referência à **folha de horas** à qual o registo corresponde.
- Tarefa – lista que guarda tarefas de aprovação de folhas de horas e de folhas de alocação interna. Os itens desta lista contêm:
  - Um **Aprovador** que é um colaborador responsável por executar a tarefa;

- Um **Colaborador**, se aplicável, que foi responsável pelo início da tarefa;
- Uma **Descrição** que define o tipo de tarefa;
- Uma referência ao **Período Semanal**, se aplicável, ao qual a tarefa pertence;
- Uma referência ao **Período Mensal** ao qual a tarefa pertence.
- Folha de Alocação Interna – lista que cria o conceito de folha em que os responsáveis de cada estrutura podem distribuir as horas alocadas à sua estrutura em serviços finais. Os itens desta lista contêm:
  - Um **Estado** que pode ser “Aberto”, “Fechado” ou “Em Falta”;
  - Uma **Organização**, ou seja, uma estrutura à qual a folha pertence;
  - Uma referência ao **Período Mensal** ao qual a distribuição será feita.
- Linha de Alocação Interna – lista que guarda as alocações internas, por estrutura, que terão de ser distribuídas pelos responsáveis das estruturas. Os itens desta lista contêm:
  - Um **Valor** total que terá de ser distribuído;
  - Uma **Organização** que foi responsável pela atribuição de horas;
  - Uma referência a um **Período Semanal** da alocação de horas;
  - Uma referência ao **Serviço** que foi alocado;
  - Uma referência à **Folha de Alocação Interna** à qual a linha pertence.
- Distribuição de Alocação Interna – lista que guarda as distribuições, das horas alocadas internamente à instituição, para serviços finais. Os itens desta lista contêm:
  - Um **Valor** das horas distribuídas do valor total da linha de alocação interna correspondente;
  - Uma referência a um **Serviço** para o qual a distribuição foi aplicada;
  - Uma referência a uma **Linha de Alocação Interna** à qual a distribuição pertence.
- Colaboradores – lista que guarda todos os utilizadores que podem utilizar a solução de reporte de horas, o seu aprovador e seu perfil dentro da solução. Os itens desta lista contêm:
  - Um valor booleano que determina se o utilizador está **Activo** ou não;

- Um **Aprovador** que será um colaborador responsável por aprovar as horas deste colaborador;
- Uma **Organização** à qual o colaborador pertence;
- Um **Número de Colaborador**;
- Um **Código de Colaborador**;
- Um **Perfil** que poderá ser “Colaborador”, “Director” ou “Manager”;
- Um **Colaborador** a quem o registo pertence.
- **Catalogo de Serviços** – lista que guarda todos os serviços que cada estrutura da instituição fornece, sendo que esta definição de serviços prestados pode também ser definido ao nível das diferentes áreas de uma estrutura. Os itens desta lista contêm:
  - Um valor booleano que determina se o valor está **Activo** ou não;
  - Uma **Descrição** do serviço;
  - Uma **Organização** à qual o serviço está atribuído;
  - Um **Tipo de Serviço** que pode ser “Normal” ou “Ausência”;
  - Uma **Data de Início**, data a partir da qual o serviço passa a estar disponível;
  - Uma **Data de Fim**, data a partir da qual o serviço deixa de estar disponível;
  - Um **Nome** do serviço.
- **Delegação** – lista que guarda as delegações de permissões dentro da solução. Os itens desta lista contêm:
  - Um **Aprovador** que é um colaborador que irá delegar as suas permissões;
  - Um **Delegado** que é um colaborador que irá receber as permissões do aprovador;
  - Uma **Organização** sobre a qual as permissões de acesso serão aplicadas.

Este diagrama oculta as colunas criadas automaticamente pelo Sharepoint, sendo eles: “Criado por”, “Modificado”, “Id”, “Modificado por”, “Título”.

A Taxonomia disponibiliza dois conjuntos de termos diferentes.

O primeiro é o conjunto de termos dos Clientes que define todos os clientes da instituição. Cada termo deste conjunto contém os seguintes identificadores:

- **Código** - define o cliente;

- **Tipologia** - pode ser “Cliente” ou “Agrupamento de Clientes”.

O segundo é o conjunto de termos da organização que define a estrutura da instituição em estruturas e áreas dessa estrutura se aplicável. Cada termo deste conjunto contém os seguintes identificadores:

- **Centro de custo;**
- **Código de área;**
- **Estrutura** – um valor booleano que determina se o termo representa uma estrutura ou uma área de uma estrutura;
- **Nome da Estrutura** – identifica a estrutura que está a ser definida;
- **Sigla** – abreviatura do nome da estrutura;
- **Sigla pai** - no caso de o termo definir uma área de uma estrutura este valor será o correspondente à sigla da estrutura, no caso do termo ser uma estrutura, este será igual à **Sigla**.

### 3.3 Comunicação entre o *front-end* e o Sharepoint

A solução apenas usa comunicações dentro do Sharepoint para realizar todas as operações necessárias, não recorrendo a serviços externos a este. A Figura 8 representa os pedidos CSOM.

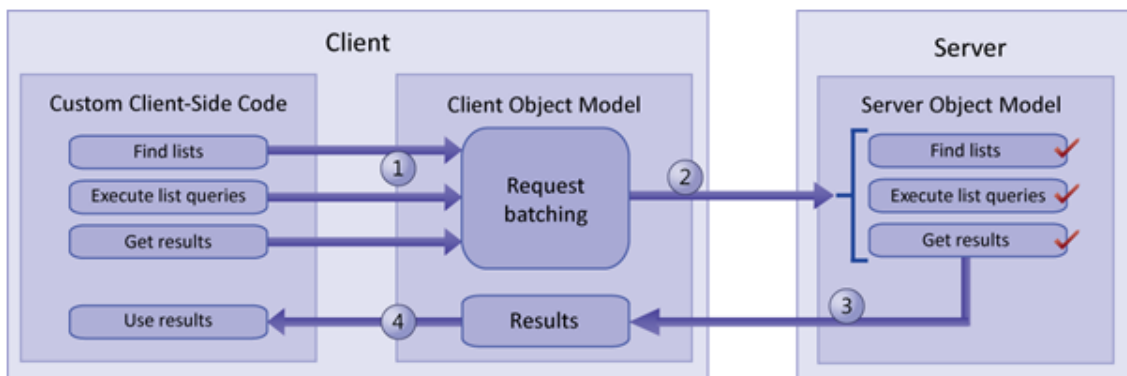


Figura 8- modelo de pedido CSOM

O pedido começa por ser construído do lado do cliente, com recurso à biblioteca de JavaScript do Sharepoint. Na Figura 9 podemos ver um exemplo do código necessário para realizar o primeiro passo para realizar a operação de atualizar um item numa lista.



```

41      var clientContext = new SP.ClientContext("endereço do site");
42
43      var web = clientContext.get_web();
44      var webList = web.getList("endereço da lista");
45
46      var listItem = webList.getItemById(inFolha.Id);
47
48      listItem.set_item('Editavel', inFolha.Activo);
49      listItem.set_item('Estado', inFolha.EstadoWorkflow);
50      listItem.set_item('Motivo', inFolha.Motivo);
51
52      listItem.update();

```

*Figura 9 - exemplo da criação de pedido CSOM*

Na linha 41 obtém-se o contexto da operação, neste caso, o site onde se encontra a lista. Nas linhas 43 e 44 obtém-se a lista e na linha 46 é obtido um item da lista a partir de um identificador. Após obter este item da lista, serão realizadas 3 operações de atualização do item, linha 48 à linha 50. O último passo da construção do pedido é a instrução para atualização do item, na linha 52, sem esta instrução o item não é atualizado na lista.

```

54      clientContext.executeQueryAsync(function () {
55
56          if (onLoad)
57              onLoad(inFolha);
58
59      }, onError);

```

*Figura 10 - exemplo de execução e utilização da resposta de um pedido CSOM*

Na Figura 10 acima vemos o código necessário para os passos 2 e 4 da Figura 8. Na linha 54 é chamada uma função que executa o pedido assincronamente, isto significa que o cliente continua a executar sem bloquear à espera de uma resposta do servidor. O passo 3 não ocorre do lado do cliente, mas sim do lado do servidor e devido à tipologia do Sharepoint utilizado nesta solução, a execução do pedido é feita nos servidores da Microsoft. Quando o pedido for resolvido e retornada uma resposta, há dois caminhos de execução possíveis- Se o pedido falhar por algum motivo é executada a função passada no segundo parâmetro, mas se tiver sucesso é executada a função passada no primeiro parâmetro da função.

### 3.4 *Front-end*

Nesta secção são descritas as páginas criadas com recurso às tecnologias de desenvolvimento para a *web* (Html, Css e JavaScript). A criação de páginas no Sharepoint é feita através da incorporação de peças *web*, que são blocos de informação. Um exemplo de peças disponibilizadas pelo Sharepoint pode ser encontrado na Figura

11, sendo que o “editor de Scripts” foi usado para injetar os *scripts* e o HTML nas páginas.

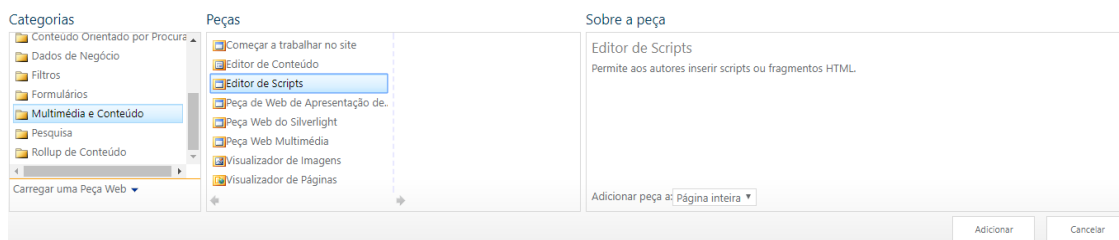


Figura 11 - Peças Web do Sharepoint

Para adicionar o HTML à página é necessário adicioná-lo ao editor de *scripts*. As páginas que serão apresentadas de seguida seguem a mesma estrutura para construir a página.

Na Figura 12 está representado um *template* da estrutura do HTML para construir as páginas. Na primeira secção, linhas 3 e 5, são importados os ficheiros CSS e bibliotecas JavaScript, na segunda secção, da linha 7 à 9, é adicionado o HTML da página e na última secção, a partir da linha 12, são carregados os *scripts* que serão usados na página e executada uma função para inicializar a página.

```

3 <link href="{endereço da página}/{Nome da lista}/{nome da pasta}/{nome do ficheiro}" rel="stylesheet" />
4
5 <script src="{endereço da página}/{Nome da lista}/{nome da pasta}/{nome do ficheiro}"></script>
6
7 <div>
8   "O Html é adicionado aqui!"
9 </div>
10
11
12 <script>
13   Common.InitSPScripts(function () {
14
15     Common.PushScript("Structures");
16     Common.PushScript("Metadata");
17     Common.PushScript("Colaboradores");
18     Common.PushScript("Periodos");
19     Common.PushScript("FolhaHoras");
20     Common.PushScript("Tarefas");
21     Common.PushScript("Delegacao");
22     Common.PushScript("PaginaInicialUI");
23
24     Common.LoadScripts(function () {
25
26       PaginaInicialUI.InitUI();
27
28     });
29   });
30
31 </script>

```

Figura 12 - template de html para páginas

A Figura 13 representa um *template* para a construção dos *scripts* de JavaScript usados nas páginas. Na variável “PaginaInicialUI” será guardado um objeto JSON com a propriedade “InitUI”. Nesta propriedade estará armazenada uma função e para invocar essa função executa-se a seguinte operação “PaginaInicialUI.InitUI()”. Ainda na mesma

figura está definida uma função privada que apenas poderá ser invocada por funções definidas neste objeto JSON.

```
2  var PaginaInicialUI = (function () {
3
4      // Private
5
6      var privateVar = "";
7
8      function privateFunction(arg1,arg2){ ...
18  }
19
20      // Public
21      var Module = {};
22
23
24      Module.InitUI = function () {
25
26          //do something
27      };
28
29      return Module;
30  }());
```

*Figura 13 - Template de um ficheiro JavaScript*

### 3.4.1 Página de gestão de períodos

A página de gestão de períodos, Figura 14, tem como objetivo permitir que o administrador funcional realize as seguintes operações:

- Abertura e fecho de períodos anuais;
- Abertura, validação e fecho de períodos mensais;
- Consultar o estado dos diferentes períodos anuais, mensais e semanais.

Esta está dividida em três secções. Na primeira está toda a informação relacionada com o período anual. A segunda contém a lista de períodos mensais com o estado correspondente e uma ação possível. Quando se clica numa linha desta secção é disponibilizada uma lista com os períodos semanais do mês selecionado. Na última secção não é possível realizar ações sobre os períodos porque estas são realizadas por um serviço instalado na nuvem da Microsoft.

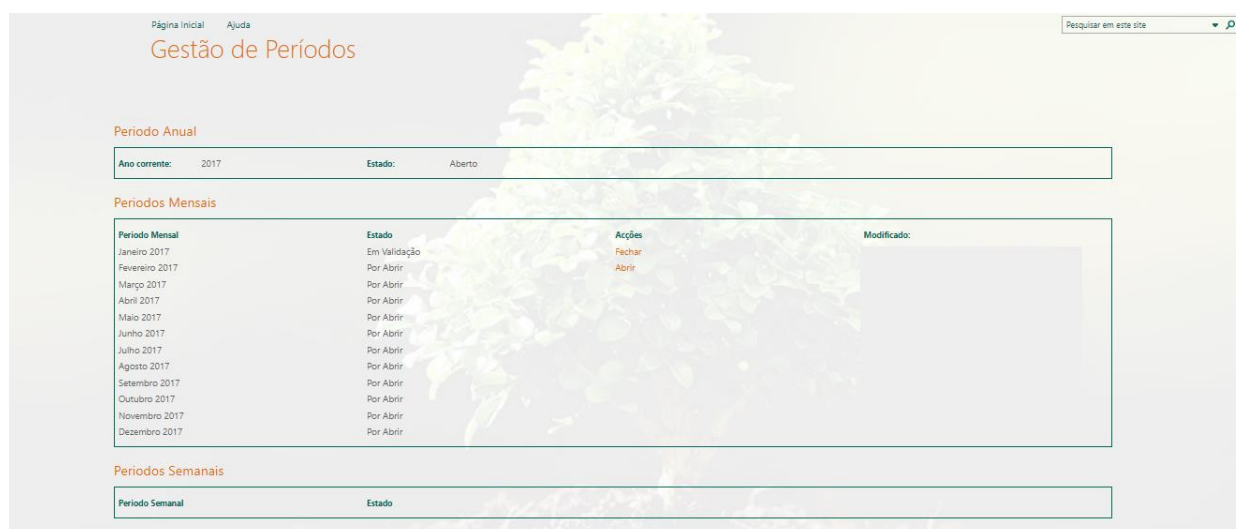


Figura 14 - Página de gestão de períodos

Após a página ser totalmente carregada, o *script* realiza um pedido ao Sharepoint para determinar se há algum período anual aberto. Se existir, volta a ser feito um pedido para obter todos os períodos mensais e semanais do ano corrente. Se não existir um período anual aberto, o script disponibiliza, na primeira secção da página um botão, cuja ação é iniciar o processo de criação de um período anual. Este serviço será responsável por criar os itens nas diferentes listas. Quando esta validação inicial está concluída, o utilizador tem a possibilidade de executar três operações sobre os períodos mensais e uma sobre os anuais. Estas ações, tal como a ação de abrir o período anual, geram tarefas que serão executas pelo serviço instalado no Azure, com a diferença que estas tarefas apenas modificam os estados dos itens para refletir as ações do utilizador. A Figura 15 ilustra o diagrama de atividades desta página.

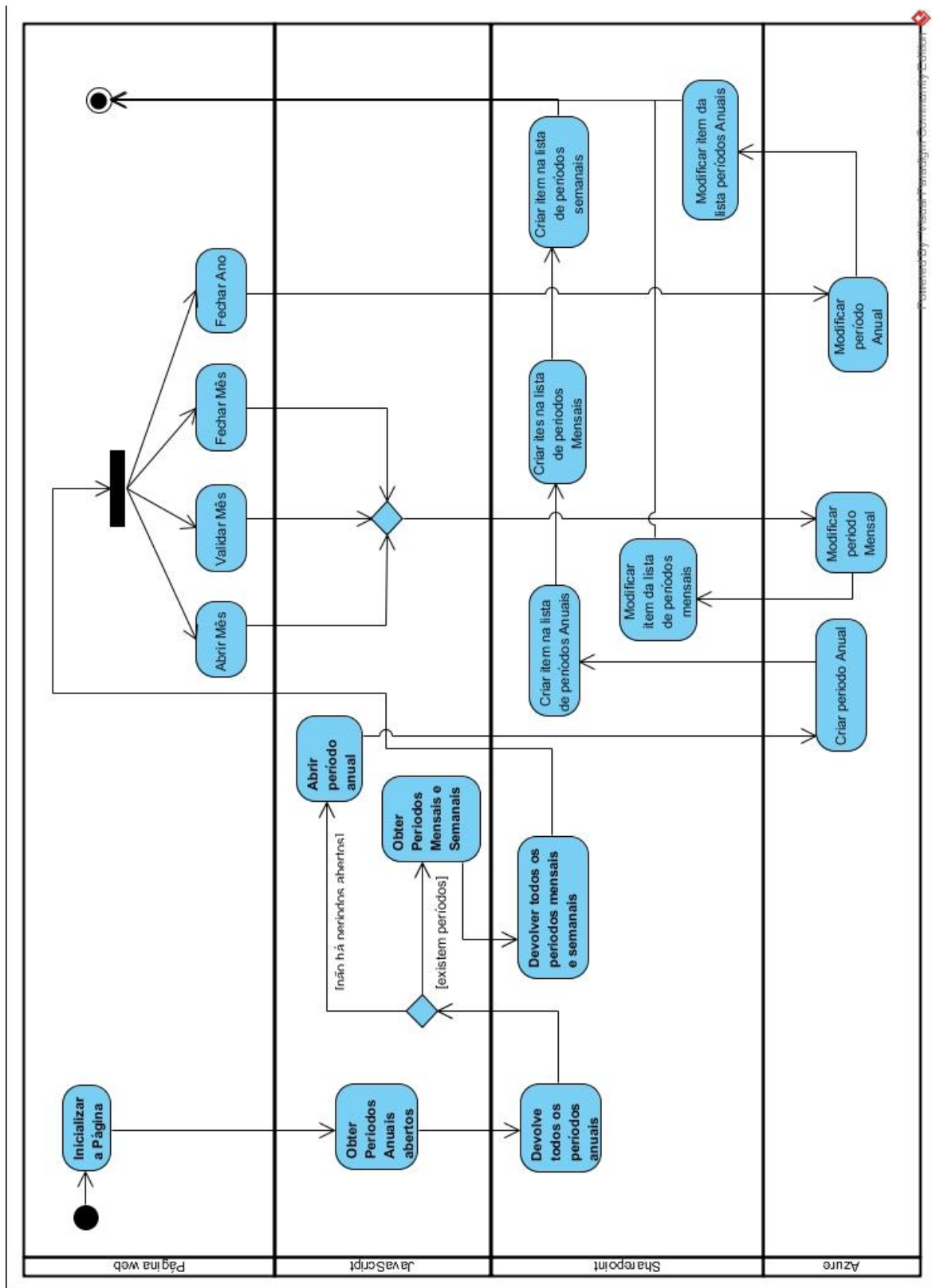


Figura 15 - Diagrama de atividades da página de gestão de períodos

### 3.4.2 Página inicial

A página inicial da solução, Figura 16, tem como objetivo apresentar o estado geral das folhas de horas e das tarefas do utilizador autenticado. O estado geral é obtido através do cálculo de indicadores, tendo como base os itens que são obtidos da lista de folhas de horas e da lista de tarefas, no Sharepoint.

Nesta página é possível realizar as seguintes operações:

- Selecionar uma folha de horas;
- Selecionar uma tarefa que ira direcionar o utilizador para uma folha de horas ou de alocação interna.



Figura 16 - Página inicial da solução

Após a página estar totalmente carregada, o *script* realiza dois pedidos ao Sharepoint para obter as folhas de horas dos períodos mensais em estado aberto, e todas as tarefas do utilizador. Quando os dados são retornados, o script calcula os indicadores que aparecem no início da página. Concluída a inicialização da página, o utilizador pode selecionar tanto uma folha como uma tarefa, sendo que ambas irão re-direcionar para uma folha de alocação.

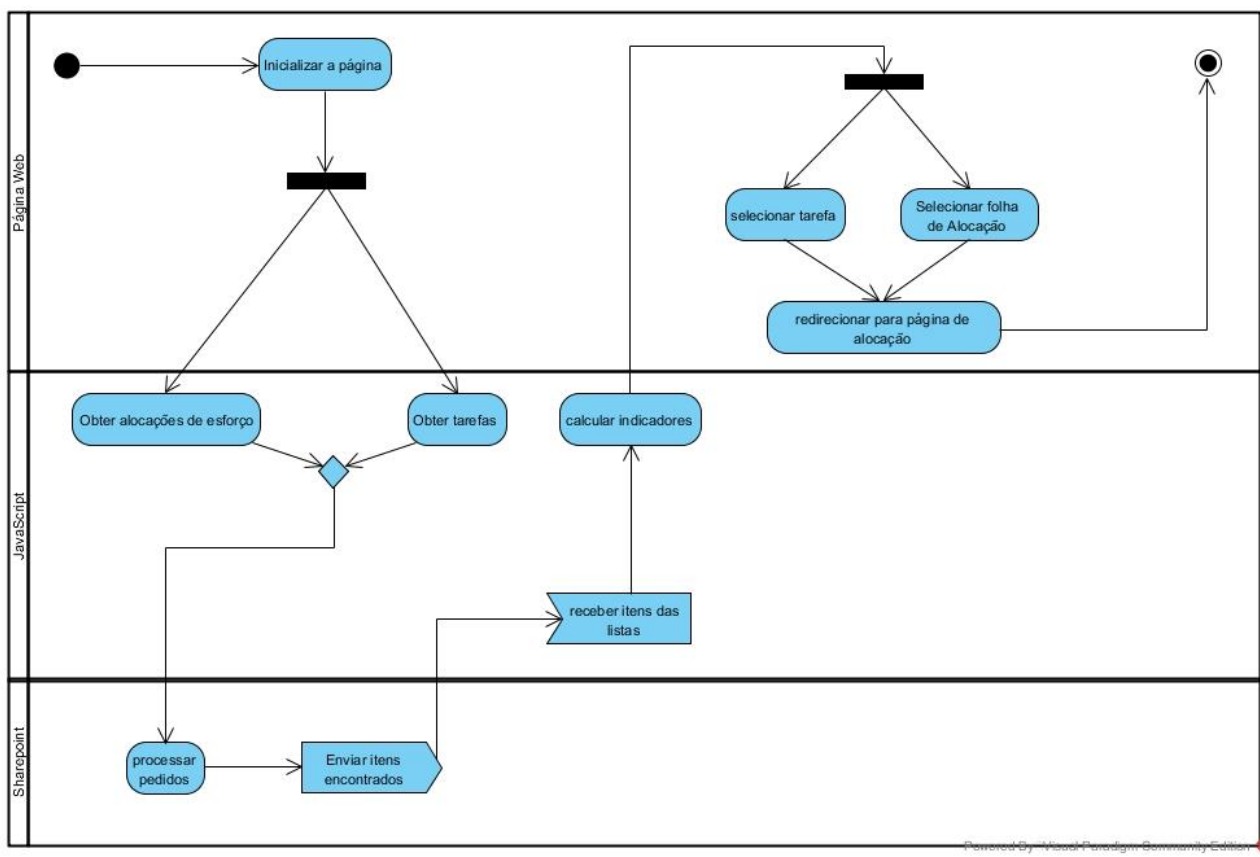


Figura 17 - Diagrama de atividades da página inicial

### 3.4.3 Páginas de listagem de informações

Nesta secção são descritas três páginas diferentes com um funcionamento muito semelhante sendo que a maior diferença é a informação que apresentam.

A página de listagem de folhas, Figura 18, permite que os utilizadores façam pesquisas às suas folhas ou no caso de serem responsáveis procurarem por folhas de outros utilizadores.

A página de listagem de tarefas, Figura 19, permite que os utilizadores façam pesquisas relativamente às tarefas dos diferentes utilizadores.

A página de relatórios, Figura 20, permite que os utilizadores façam pesquisas relativamente a clientes e serviços, em um ou mais períodos semanais, para posteriormente extrair um ficheiro que poderá ser aberto com o Microsoft Excel.

Cada página está dividida em duas secções. A primeira secção disponibiliza aos utilizadores um conjunto de caixas que permite seleccionar: a estrutura; a área de uma estrutura; colaboradores da área seleccionada; período semanal; entre outros. A segunda secção disponibiliza os resultados da pesquisa efetuada na secção anterior, em forma de

lista. No caso da Figura 18 a secção de resultados apresenta todas as folhas do utilizador que está autenticado, pois ainda não foi executada nenhuma pesquisa.

No diagrama de atividades da página de listagem de folhas de horas, Figura 21, vemos que para a inicializar a mesma são executados diferentes pedidos. Estes pedidos têm como objetivo obter os dados necessários para preencher as caixas de pesquisa e obter a listagem de folhas do utilizador. Após estes serem enviados pelo Sharepoint e adicionados à página, o utilizador pode realizar uma das seguintes ações; ou escolhe um resultado “clicando” na linha correspondente ou seleciona um conjunto de filtros e carrega no botão “Pesquisar”, o que irá criar um pedido com os filtros selecionados e executa-lo.



## Alocação de Esforço

Pesquisar Folhas de Alocação Por:

Estruturas:

Período Semanal:

Semana de 02/01 a 06/01

Áreas:

Ano:

2017

Colaboradores:

Pesquisar

### Alocações de Esforço

Colaborador	Período	Estado	Motivo	Modificado
Rosário	Semana de 30/01 a 31/01	Aprovado	-	18/01/2017
Rosário	Semana de 01/06 a 02/06	Por iniciar	-	04/02/2017
Rosário	Semana de 05/06 a 09/06	Por iniciar	-	04/02/2017
Rosário	Semana de 12/06 a 16/06	Por iniciar	-	04/02/2017
Rosário	Semana de 19/06 a 23/06	Por iniciar	-	04/02/2017
Rosário	Semana de 26/06 a 30/06	Por iniciar	-	04/02/2017

Figura 18 - Página de listagem de folhas de horas

## Listagem de tarefas

Pesquisar Tarefas Por:

Estruturas:	<input type="text" value="Assessoria"/>	Período Mensal:	<input type="text" value="Janeiro 2016"/>
Áreas:	<input type="text" value="Assessoria"/>	<input type="button" value="Pesquisar"/>	
Colaboradores:	<input type="text" value="Coordenador DO1"/>		

### Lista das suas tarefas

Tarefa	Período	Colaborador	Assignado a
Aprovar folha de alocação	Maio 2016	Colaborador DO1	Coordenador DO1

Figura 19 - Página de listagem de tarefas

lade [Página Inicial](#) [Alocação de Esforço](#) [Tarefas](#) [Alocação Interna](#) **Relatórios** [Ajuda](#) [Administração](#)

## Relatórios

Pesquisar Alocações Diárias Por:

Estruturas:	<input type="text" value=""/>	Períodos Semanais:	<input type="text" value="Semana de 02/01 a 06/01"/>
Áreas:		Serviço:	<input type="text" value="Gestão"/>
Colaboradores:	<input type="text" value=""/>	Cliente:	<input type="text" value=""/>
<input type="button" value="Pesquisar"/>			

### Alocações Diárias

Colaborador	Data	Serviço	Cliente
-------------	------	---------	---------

Figura 20 - Página de relatórios

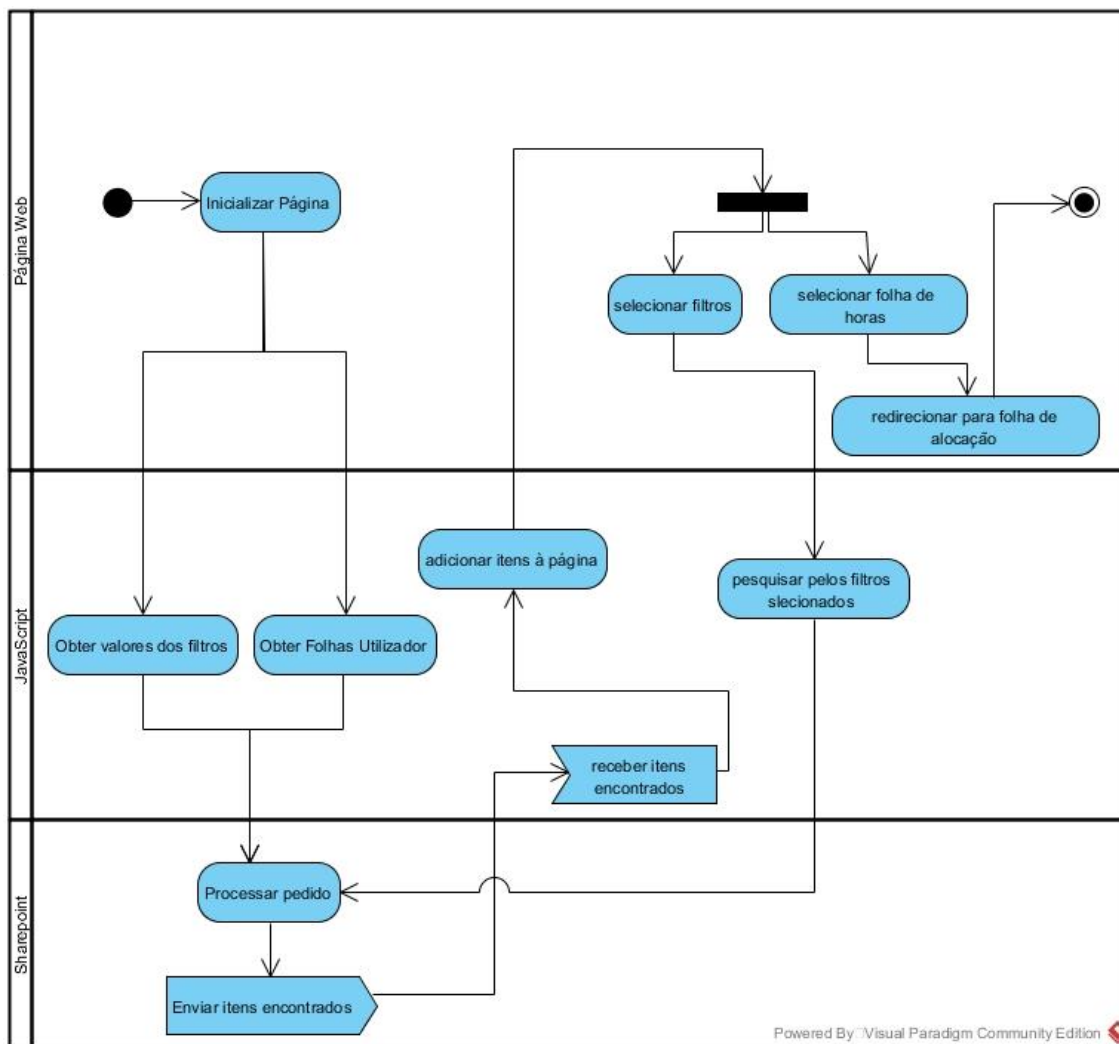





Figura 21 - Diagrama de atividades da página de listagem de folhas de horas

### 3.4.4 Folha de Horas

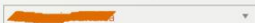














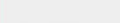



A página da folha de horas, Figura 22, serve para os utilizadores da solução adicionarem registos das horas que gastaram a realizar um serviço para um determinado cliente. Para o cliente, esse registo de horas é feito em percentagem, que mais tarde é convertida em horas, sendo que cem por cento equivale a sete horas de trabalho. Esta página está dividida em três secções. No topo, é apresentada informação sobre o titular da folha e a que estrutura este pertence, o período anual, mensal e semanal e o estado em que se encontra. A segunda secção é onde o utilizador adiciona registos, escolhendo um serviço e um cliente e nas colunas dos dias do período semanal introduz as horas com correspondência em percentagem. Para criar um registo, o utilizador tem de carregar no “+” que se encontra no fim da primeira linha. Quando for criado o registo, este é adicionado à lista de registos com um símbolo verde, o que indica que o registo não foi gravado. Apenas antes dos registos serem gravados, através do botão “Gravar”,

podem ser adicionadas notas a esse registro. Para submeter a folha de horas, o utilizador necessita de ter todos os dias do período semanal com cem por cento atribuídos e todos os registos gravados. Quando a folha está em estado submetido, os botões “Gravar” e “Submeter” são substituídos pelos botões “Aprovar” e “Rejeitar”. O responsável por aprovar a folha é notificado, através de uma mensagem de correio eletrónico, quando a folha é submetida e pode então aprovar a folha e o ciclo termina ou pode rejeitar, o que faz com que a folha volte a estar aberta para ser editada e submetida novamente. Na última secção podem ser visualizadas todas as ações sobre a folha, nomeadamente a ação realizada, a data, quem executou a ação e qual foi o estado resultante da ação. No diagrama de atividades desta página, Figura 23, estão descritas as diferentes interações que o utilizador pode realizar nesta página.

Estrutura:   
Área:  (DAJ)  
Colaborador:  Fernandes

Ano: 2017  
Mês: Maio 2017  
Semana: Semana de 01/05 a 05/05  
Estado: Por iniciar

### Alocação de Esforço

Serviço	Cliente	01/05 Segunda	02/05 Terça	03/05 Quarta	04/05 Quinta	05/05 Sexta	
	 Antas	<input type="text" value="25"/> %	<input type="text" value="25"/> %	<input type="text" value="25"/> %	<input type="text" value="25"/> %	<input type="text" value="25"/> %	
	 Antas	25,00%	25,00%	25,00%	25,00%	25,00%	 
	 Antas	25,00%	25,00%	25,00%	25,00%	25,00%	 
	 Antas	25,00%	25,00%	25,00%	25,00%	25,00%	 
	 Antas	25,00%	25,00%	25,00%	25,00%	25,00%	 
<b>Total:</b>		100,00%	100,00%	100,00%	100,00%	100,00%	

Gravar Submeter Cancelar

### Histórico da Folha

Colaborador	Atualizado	Estado	Motivo / Nota
-------------	------------	--------	---------------

Figura 22 - Página das folhas de horas

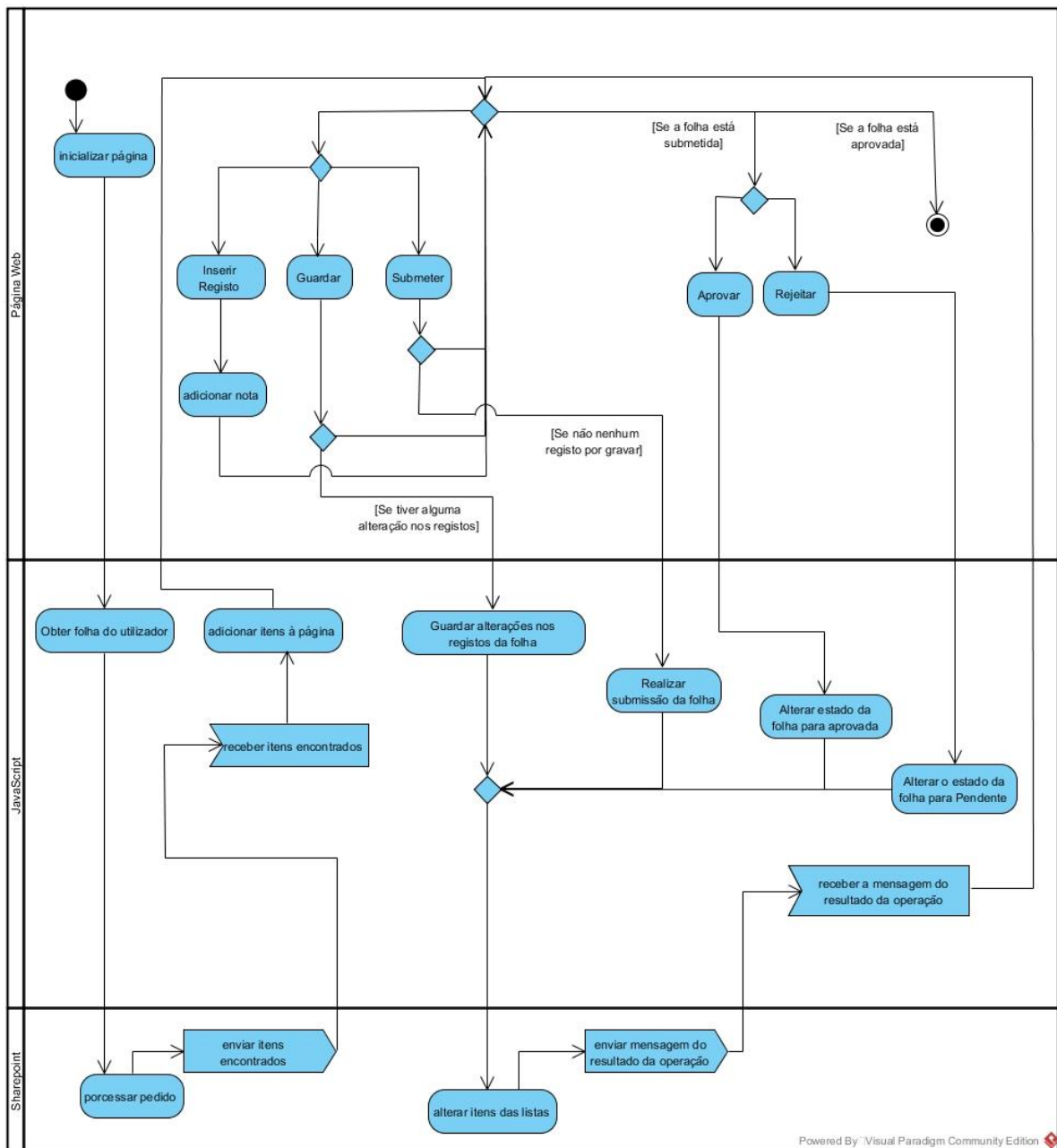


Figura 23 - Diagrama de atividades da página das folhas de horas

### 3.4.5 Página da folha de alocações internas

Segundo o modelo de negócio da instituição onde este projeto foi desenvolvido, as horas podem ser atribuídas a serviços cujo cliente é uma estrutura interna do mesmo. Uma estrutura “A” apenas atribui horas de um serviço para a estrutura “B” quando esta última o solicita. A estrutura “B” pode solicitar os serviços da estrutura “A” para realizar um serviço que presta e desta forma os custos do serviço que a estrutura “A” prestou têm de ser atribuídos ao serviço que o originou. Quando o administrador funcional da solução, na página de gestão de períodos, coloca o mês em validação, os

responsáveis das estruturas internas do cliente recebem uma folha com todos os serviços e horas que lhe foram atribuídos e com isso distribui as horas pelos serviços finais.

A página da folha de alocação interna pode ser vista na Figura 24. Esta está dividida em duas secções. Na primeira secção é possível, de acordo com o perfil, escolher uma estrutura e visualizar a lista de folhas de alocação interna, que são criadas mensalmente. Quando uma folha é selecionada, através de um clique na linha correspondente, é carregado na segunda secção da página o somatório dos diferentes serviços atribuídos a esta estrutura. Os serviços aparecem agrupados por período semanal e indicam qual a estrutura que atribui o serviço e o somatório de todos registos. Para adicionar uma distribuição é necessário clicar no botão “+” no fim de cada linha, que irá adicionar um registo logo a baixo desta com uma caixa de seleção para o utilizador seleccionar o serviço final. Para concluir o processo o utilizador deverá distribuir a totalidade das horas atribuídas, guardar e só depois submeter. O diagrama de atividades desta página pode ser encontrado na Figura 25.

**Folhas de alocação Interna**

Escolha a estrutura que pretende visualizar: [Antas]

Estrutura: Antas      Período Mensal: Janeiro 2017      Estado: Fechado

**Distribuição de alocações para Antas - Janeiro 2017**

Semana de 02/01 a 06/01

Compliance	Execução de Reaberturas do Mês (FMS)	200,00 Horas	+
	Clientes - Retalho	10,00 Horas	x
	Clientes - Retalho	190,00 Horas	x

Gravar

Figura 24 - Página da folha de alocação interna



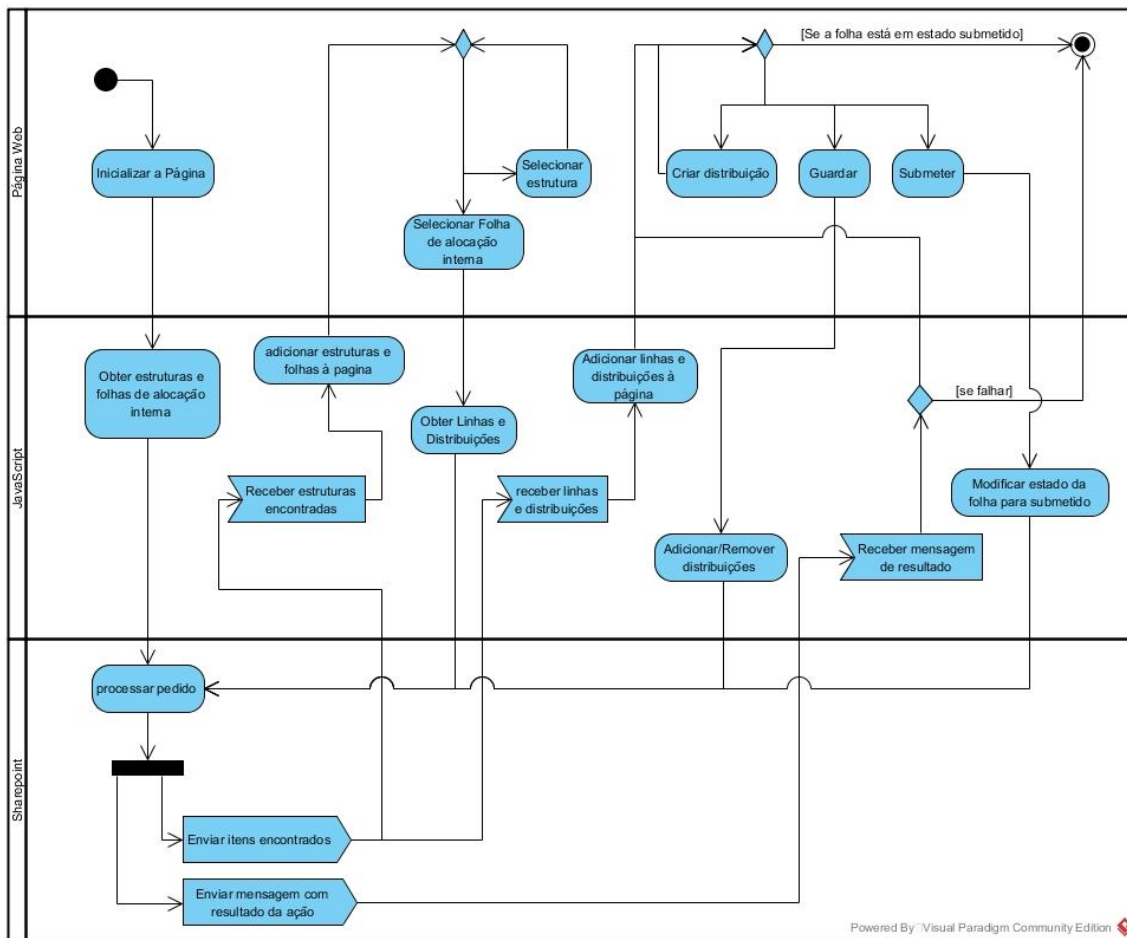


Figura 25 - Diagrama de atividades da folha de alocação interna

## 3.5 Backend

O *backend* da solução foi construído com recurso à ferramenta de trabalho do “Azure Webjobs”, o que simplifica a tarefa de criar código para processamento em *background*. Esta solução utiliza um programa escrito em C#, que na sua base interage com o Sharepoint para criar, apagar, editar e consultar itens das diferentes listas apresentadas na secção 3.2 . Através da conjugação destas ações elementares com o Sharepoint foram construídas operações necessárias ao bom funcionamento do sistema.

A cada 15 minutos, este programa executa uma função que faz um pedido CSOM com o objetivo de obter todos os elementos da lista “WorkOrders”, cujo estado é pendente. (Quando existem elementos nesta lista que satisfaçam esta condição o programa cria uma *thread* para executar essa tarefa.) Se após a execução do pedido forem retornados elementos da lista que estejam nesse estado o programa cria um *thread* para executar cada tarefa retornada.

As tarefas realizadas pelo *webjob* são: abertura e fecho de períodos anuais, mensais e semanais; validação dos períodos mensais; envio de diversas notificações aos



utilizadores, através de *e-mail*; criação de relatórios com as horas reportadas e com as distribuições internas das mesmas.

### **3.6 Entrada em produção e apoio à mudança**

A entrada em produção foi realizada após serem dadas formações a todos os utilizadores finais da solução, que se dividiram por 11 sessões em Lisboa e 3 no Porto, para um total de 500 pessoas. Adicionalmente, foram previamente criados conteúdos para esclarecimento de dúvidas, divulgação e explicações sobre o funcionamento da aplicação e os procedimentos que devem ser efetuados.

Após a entrada de formação seguiu-se um período de 2 meses de suporte de segunda linha que visava a resolução de todos os defeitos técnicos e esclarecimento de dúvidas dos utilizadores.



## Capítulo 4 Solução de *e-learning*

A solução de *e-learning* surgiu da necessidade que o cliente tinha de atualizar e melhorar a ferramenta utilizada para oferecer formação aos seus comerciais.

O ciclo de funcionamento da aplicação começa por fazer um mapeamento entre os produtos e serviços que o cliente tem com os cursos que disponibiliza aos seus colaboradores. De seguida, o sistema identifica as necessidades de formação de cada colaborador. Para identificar estas necessidades, o sistema calcula a efetividade de vendas, para de cada produto e cada colaborador. Desta forma é possível calcular o rendimento médio geral. Com o rendimento médio geral calculado, o ciclo entra na terceira fase para sugerir cursos a utilizadores. O sistema pesquisa para cada produto quem está abaixo da média geral e sugere cursos que foram mapeados na primeira fase do ciclo. Por último, o sistema mede o impacto das formações sugeridas, através de dados obtidos por outros sistemas.

Neste capítulo serão explorados os componentes do sistema que foram da minha responsabilidade desenvolver, nomeadamente, duas coleções de sites do Sharepoint. Todos os outros componentes serão abordados brevemente com o objetivo de fornecer uma visão geral da solução.

## 4.1 Arquitetura

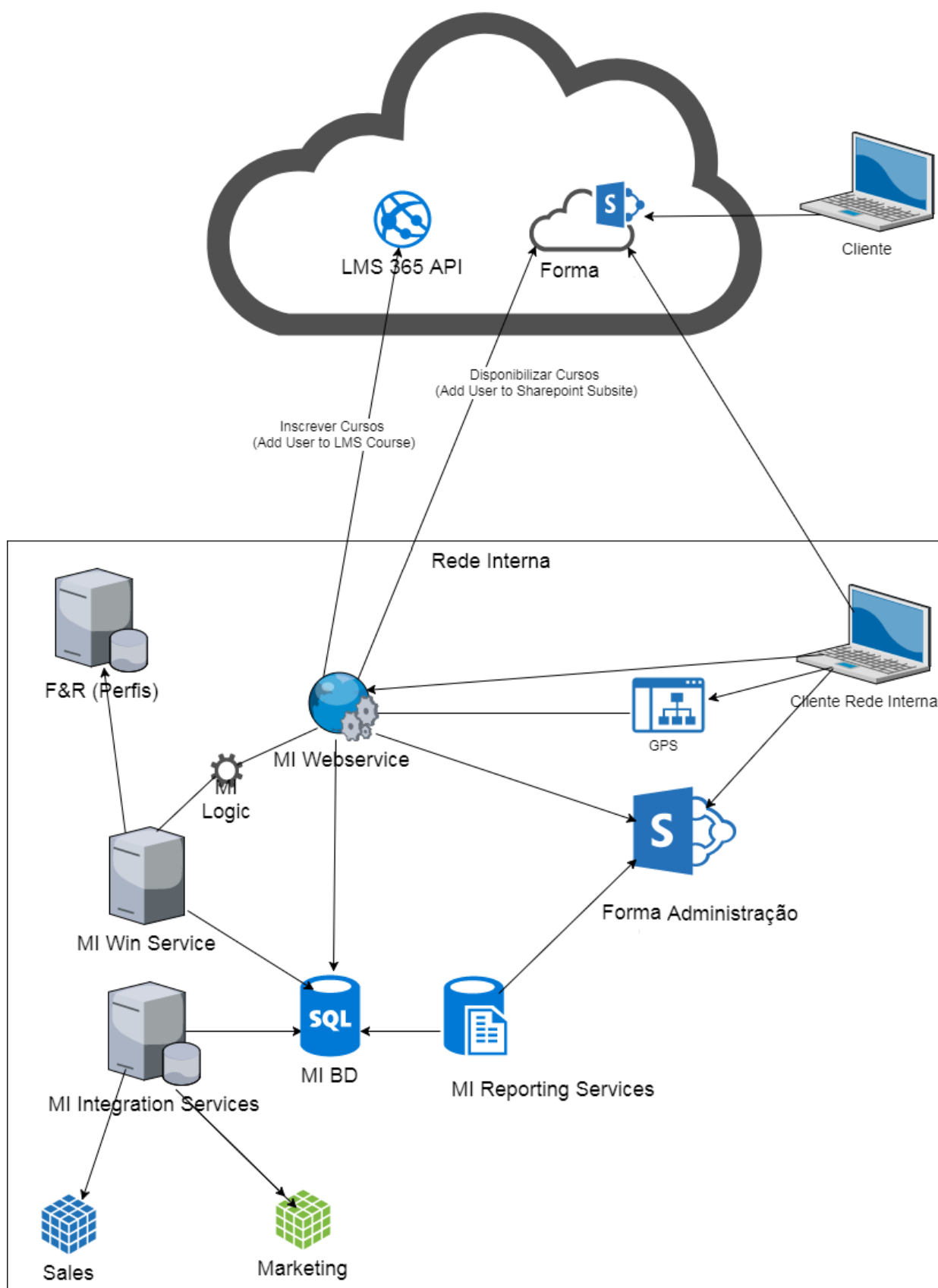


Figura 26 - Componentes da solução de e-learning

Inicialmente, a solução foi pensada para ser totalmente construída na nuvem e apenas ser integrada com sistemas hospedados pelo cliente, mas isso não era compatível com as políticas de segurança do cliente. Na nuvem apenas é feita a apresentação de dados e gestão de conteúdos, enquanto todos os dados relativos aos colaboradores e os resultados da execução das suas atividades laborais são mantidos pelo cliente nas suas instalações.

O LMS 365 (*Learn Management System*) é uma ferramenta *online*, que integra com o Sharepoint *Online*. É amplamente usada nesta solução, pois permite criar cursos e planos de aprendizagem, adicionar diferentes tipos de conteúdos aos cursos, acompanhar o progresso dos utilizadores na execução, entre outros. Nesta solução são usadas as peças *web* que esta ferramenta disponibiliza para o Sharepoint. Estas peças têm como objetivo mostrar os cursos em que o colaborador está inscrito ou em que se pode inscrever e qual o seu progresso.

Na Figura 26 estão ilustrados os componentes que constituem a solução.

Numa representação de nuvem estão o “LMS 365 API” e o “Forma”. O primeiro consiste num serviço *web RESTful*, do LMS 365, que permite fazer a gestão de cursos e de conteúdos. O segundo consiste numa coleção de *sites* chamada “Forma” onde são criados *sub-sites* para cada curso.

Na rede interna do cliente existem diversos componentes que permitem a recolha de dados e fazem a seleção de cursos a serem sugeridos para os colaboradores. Um dos componentes é uma coleção de *sites*, em Sharepoint *On-premises*, na qual foram criadas todas as páginas de administração da solução. Estas páginas interagem, maioritariamente, com o serviço *web* do módulo de integração. O módulo de integração é um conjunto de serviços que unifica toda a informação relevante para o sistema na base de dados do mesmo. Na Tabela 1 - Funcionalidades do módulo de integração são enumeradas todas as funcionalidades que este módulo adiciona à solução, bem como a identificação da origem dos dados e o destinatário que irá utilizar os mesmos.

<b>Funcionalidades</b>	<b>Origem</b>	<b>Destino</b>
<b>Inscrição automática dos utilizadores em cursos através da implementação de um <i>workflow</i> de aprovação.</b>	MI	LMS
<b>Informação sobre cursos efetuados/por realizar e progresso do utilizador por campanha</b>	LMS	MI
<b>Performance do utilizador em relação aos objetivos por campanha (no GPS)</b>	GPS	MI
<b>Identificação de incumpridores na execução das formações da campanha</b>	MI	GPS
<b>Identificação utilizadores que necessitam de executar cursos devido a maus resultados</b>	MI	LMS, GPS
<b>Identificação de utilizadores que pertencem a campanhas novas</b>	GPS	MI
<b>Identificação Novas Campanhas</b>	GPS	MI
<b>Mapeamento Campanhas para Curso</b>	GPS	MI
<b>Relatório Formações concluídas/não, concluídas vs. desempenho do utilizador</b>	MI	GPS
<b>Relatório Semanal de Progresso/Avaliação Utilizadores nos vários cursos</b>	MI	GPS
<b>Identificação de certificados a emitir pelo SIGO</b>	MI	SIGO
<b>Identificação de utilizadores no SIGO</b>	MI	SIGO
<b>Extração de listagem de certificados a pedir à plataforma SIGO (utilizadores, curso, data, nota) e envio para plataforma</b>	MI	SIGO

*Tabela 1 - Funcionalidades do módulo de integração*

### 4.1.1 Estrutura da coleção de sites “Forma”

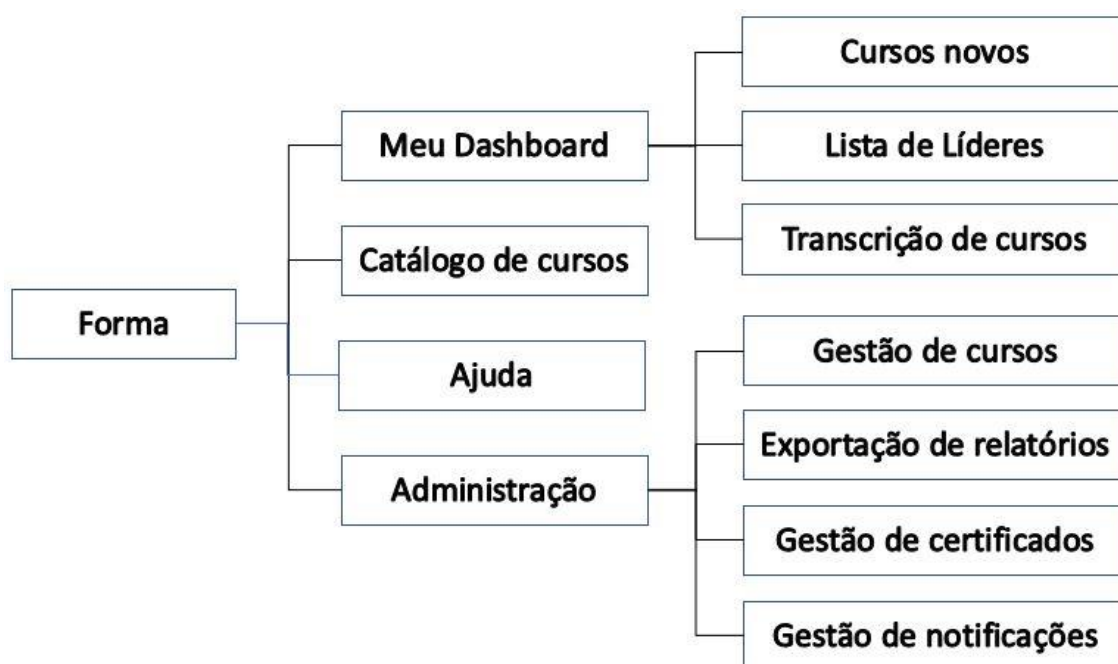


Figura 27 - Estrutura da coleção de sites "Forma"

A coleção de *sites* “Forma”, criada no Sharepoint Online, é constituída por três *sites*: *Meu Dashboard*, *Catálogo de cursos* e *Ajuda*. A “Administração” para fazer a gestão de cursos, exportação de relatórios, gestão de certificados e notificações é realizada numa página criada do LMS 365 num ambiente externo ao Sharepoint Online do cliente. A sua representação na Figura 27 corresponde a um *link* na barra de navegação das coleções de *sites*, que aponta para uma página neste ambiente externo à solução.

O “Meu Dashboard” consiste numa página com indicadores sobre o número de cursos que foram sugeridos ao utilizador, no trimestre atual, e pontos alcançados com a realização dos cursos durante o trimestre e o ano civil atuais. Por baixo destes indicadores foi adicionada uma peça *web*, fornecida pelo LMS 365, que lista todos os cursos que o utilizador está inscrito, a realizar ou já realizou, uma lista de utilizadores com mais pontos obtidos, entre outros indicadores. A Figura 28 é um exemplo retirado do manual da ferramenta e mostra as diferentes informações que esta peça pode fornecer.

My Training Dashboard			
<b>Anna Ageucheva</b>		<b>5 / 10</b> Courses Completed	<b>4</b> in your company ranking
		<b>253</b> CEUs	
<a href="#">Current Training</a> <a href="#">In Progress</a> <a href="#">Completed</a> <a href="#">Pending</a> <a href="#">Certificates</a> <a href="#">CEUs</a> <a href="#">Leaderboard</a>			
Course Name ^	Due Date	Progress	Actions
<b>Business and Management Course</b> e-Learning		Not Started	
<b>Hardware Course</b> e-Learning	11/23/2016	In Progress	
<b>Workplace Safety and Health</b> e-Learning		Not Started	

Figura 28 - Exemplo da peça web "Dashboard"

O “Catálogo de cursos” consiste numa página com uma peça web, também fornecida pelo LMS 365, que permite ao utilizador:

- pesquisar cursos aplicando filtros de texto, categoria, tipo de curso e data (lado esquerdo da Figura 29);
- visualizar informações mais detalhadas dos cursos e ingressar nos mesmos (lado direito da Figura 29).

Figura 29 - Catálogo de cursos

A “Ajuda” tem como objetivo disponibilizar manuais de utilização da ferramenta e uma lista com respostas a perguntas frequentes sobre a mesma. A estrutura da página contém duas peças web, que apontam para listas deste site onde é guardada essa informação.



#### 4.1.2 Estrutura da coleção de sites “Forma Administração”

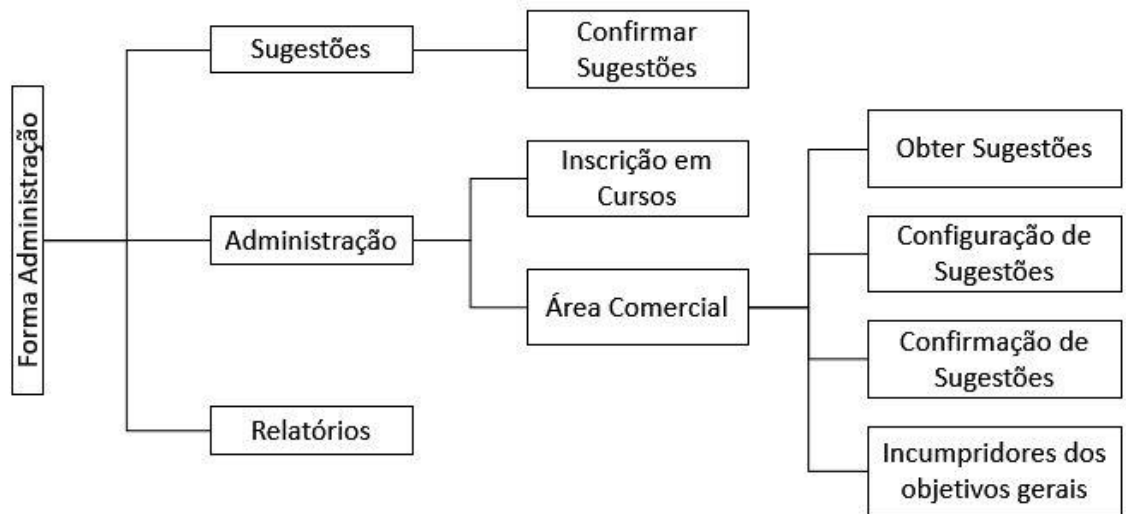


Figura 30 - Estrutura da coleção de sites "Forma Administração"

A coleção de sites “Forma Administração”, Figura 30, está dividida em três sites diferentes: “Sugestões”; “Administração”; “Relatórios”.

O site de “Relatórios” destina-se a receber páginas com diferentes relatórios. Os relatórios a ser adicionados neste site são, por exemplo, um relatório de impacto dos cursos no rendimento dos utilizadores.

O site de “Sugestões” destina-se a ser utilizado por todos os responsáveis das diferentes equipas de colaboradores. Neste site poderão confirmar as sugestões de cursos que o sistema fez para a sua equipa.

O site de “Administração” permite ao administrador funcional inscrever um conjunto de utilizadores num curso, configurar e iniciar o processo de obtenção de sugestões, confirmar as sugestões feitas pelo sistema e identificar os colaboradores que estão abaixo da média geral.

A Figura 31, tal como na secção 3.2.1 , ilustra o conjunto de ficheiros HTML, CSS e JavaScript. Os ficheiros de HTML nesta figura estão separados de acordo com os sites principais da coleção.

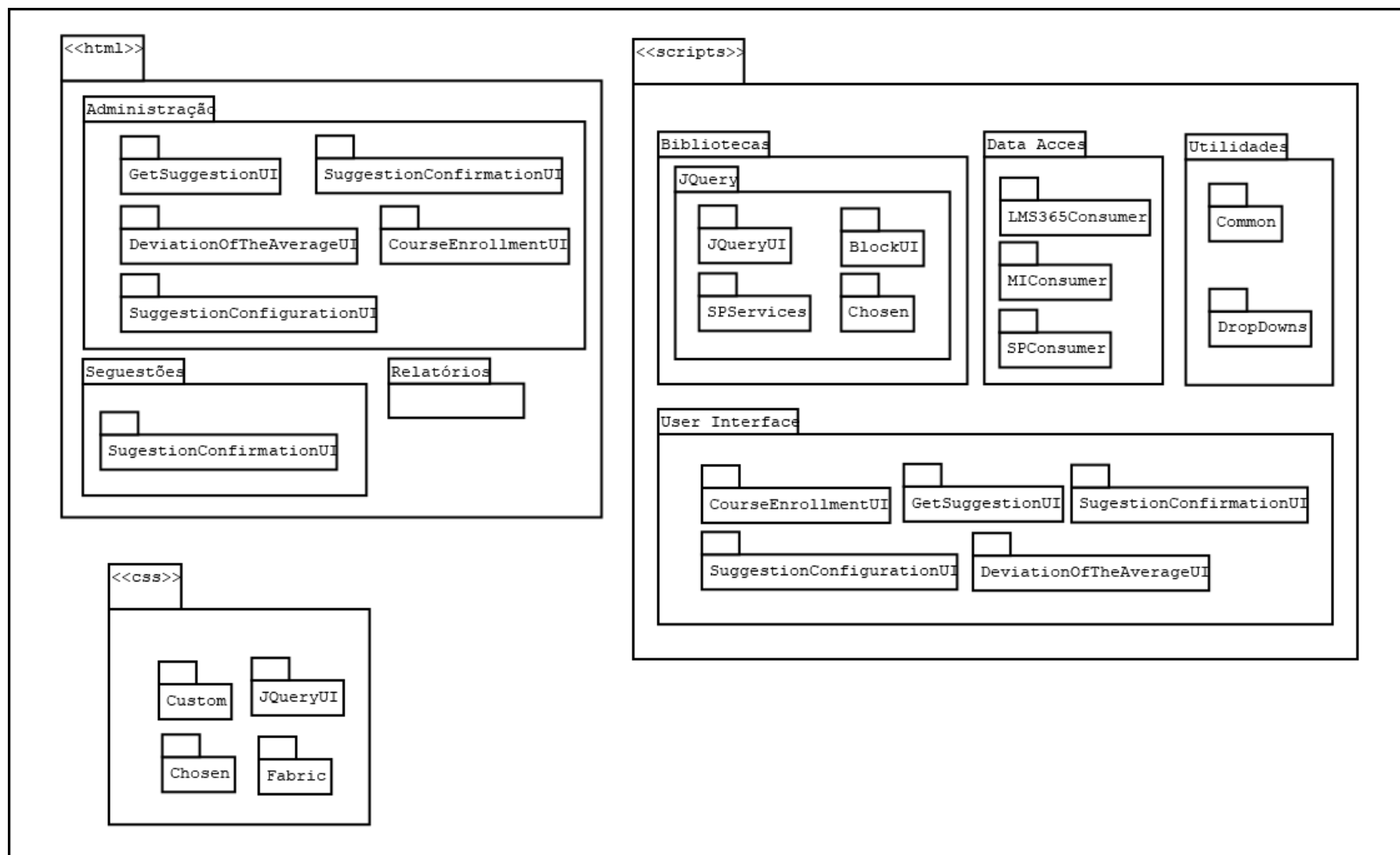


Figura 31 - Vista de módulos para a coleção de site "Forma Administração"

## 4.2 Comunicação entre o Módulo de Integração e o *Front-end*

O serviço *web* do módulo de integração é o único ponto de acesso entre o sistema montado na rede interna do cliente e o sistema montado na nuvem. Adicionalmente, por motivos de políticas de segurança do cliente, a comunicação entre a rede interna e a nuvem apenas é iniciada pelo serviço *web*. Deste modo previne-se que informação confidencial não seja extraída por sistemas externos.

A comunicação entre o *front-end* e o serviço *web* é feita através de pedidos Ajax que são construídos em JavaScript e com recurso à biblioteca JQuery. Na Figura 31 são apresentados os diferentes ficheiros utilizados para construir as páginas presentes na coleção de *sites* “Forma Administração”. Nesta secção, importa destacar o módulo “MIConsumer” que é um JavaScript no qual foram construídas todas as funções que montam o pedido Ajax. A Figura 32 mostra uma função que cria um pedido Ajax do tipo “Post”, que contém um conjunto de dados a ser utilizados pelo serviço *web*. A mesma função, quando cria o pedido, também define que funções devem ser executadas em caso de sucesso ou em caso de erro.

```
121     Module.postSubscriptionsInput = function (arrayToPost, onload, onerror) {
122
123         $.ajax({
124             type: "Post",
125             url: "http://{URL DO SERVIÇO}" + "{/URI DO SERVIÇO}",
126             data: arrayToPost,
127             contentType: "application/json",
128             success: function (response) {
129                 if (response.result.hasError) {
130                     onerror(response.result.description);
131                 } else {
132                     onload(response.users);
133                 }
134             },
135             error: onerror
136         });
137     }
```

Figura 32 - Função responsável por criar pedido Ajax

## 4.3 *Front-end*

Nesta secção são descritas apenas as páginas da coleção de *sites* “Forma Administração”, com recurso às mesmas tecnologias abordadas na secção 3.4 .

### 4.3.1 Inscrição em cursos

A página “Inscrição Cursos”, Figura 33, adiciona uma funcionalidade à ferramenta externa, LMS 365, que foi utilizada nesta solução. Na ferramenta externa, os utilizadores são adicionados aos cursos individualmente e é neste sentido que a página adiciona uma nova funcionalidade. Através dos filtros é possível pesquisar um grupo de colaboradores e inscrevê-los todos com apenas dois cliques ou, se necessário, uma parte destes seleccionando as caixas de selecção correspondentes.

The screenshot shows the 'Inscrição Cursos' page. At the top, there is a navigation bar with the logo and menu items: 'Sugestões', 'Relatórios', 'Inscrição Cursos' (active), and 'Área Comercial'. Below the navigation bar, the page title 'Inscrição Cursos' is displayed. Underneath, there is a 'Filtros' section with five filter groups: 'Cursos/Planos' (a dropdown menu), 'Perfil' (a multi-select box with 'Comercial' and 'Gestor de Recursos'), 'Estrutura' (a dropdown menu with '1280'), 'Área' (a multi-select box with 'Balcão de Bente'), and 'Estado' (a multi-select box with 'Inscrito' and 'Não inscrito'). Each filter group has a 'Limpar tudo...' link. To the right of the filters is a 'Pesquisar' button. Below the filters is a 'Resultados' section containing a table with the following data:

Utilizador ^	Perfil ^	Estrutura ^	Área ^	Estado ^	Insc. Todas	Disp. Todas
Vera De Jesus Barros Car...	Comercial	1280	Balcão de Bente	Não Inscrito	<input type="checkbox"/>	<input type="checkbox"/>

At the bottom right of the results section is a 'Confirmar' button.

Figura 33 - página de inscrição em cursos

A página divide-se em duas secções. Quando a página é inicializada, apenas a secção de filtros é disponibilizada. Nesta secção, o administrador pode seleccionar múltiplos valores em todas as caixas, exceto na primeira. Se não seleccionar um valor em pelo menos uma caixa e carregar no “Pesquisar”, será apresentada a seguinte mensagem “Introduza valores em todas as caixas de pesquisa!”. Quando o utilizador carrega no botão “Pesquisar” é efetuado um pedido ao serviço *web*, que irá obter todos os colaboradores que satisfaçam as condições dos filtros. A lista de colaboradores é apresentada na secção “Resultados”. Para cada linha da lista, o administrador funcional tem a possibilidade de inscrever ou de sugerir o curso ao colaborador. Para realizar as ações definidas o administrador carrega no botão “Confirmar”. Para inscrever um colaborador num curso, o serviço *web* adiciona-o ao grupo “Estudantes” e para sugerir um curso adiciona-o ao grupo “Visitantes”. Os grupos aos quais os colaboradores são adicionados pertencem ao *site* correspondente ao curso, na coleção de *sites* “Forma”, no Sharepoint Online. A Figura 34 apresenta o diagrama de atividades para esta página.

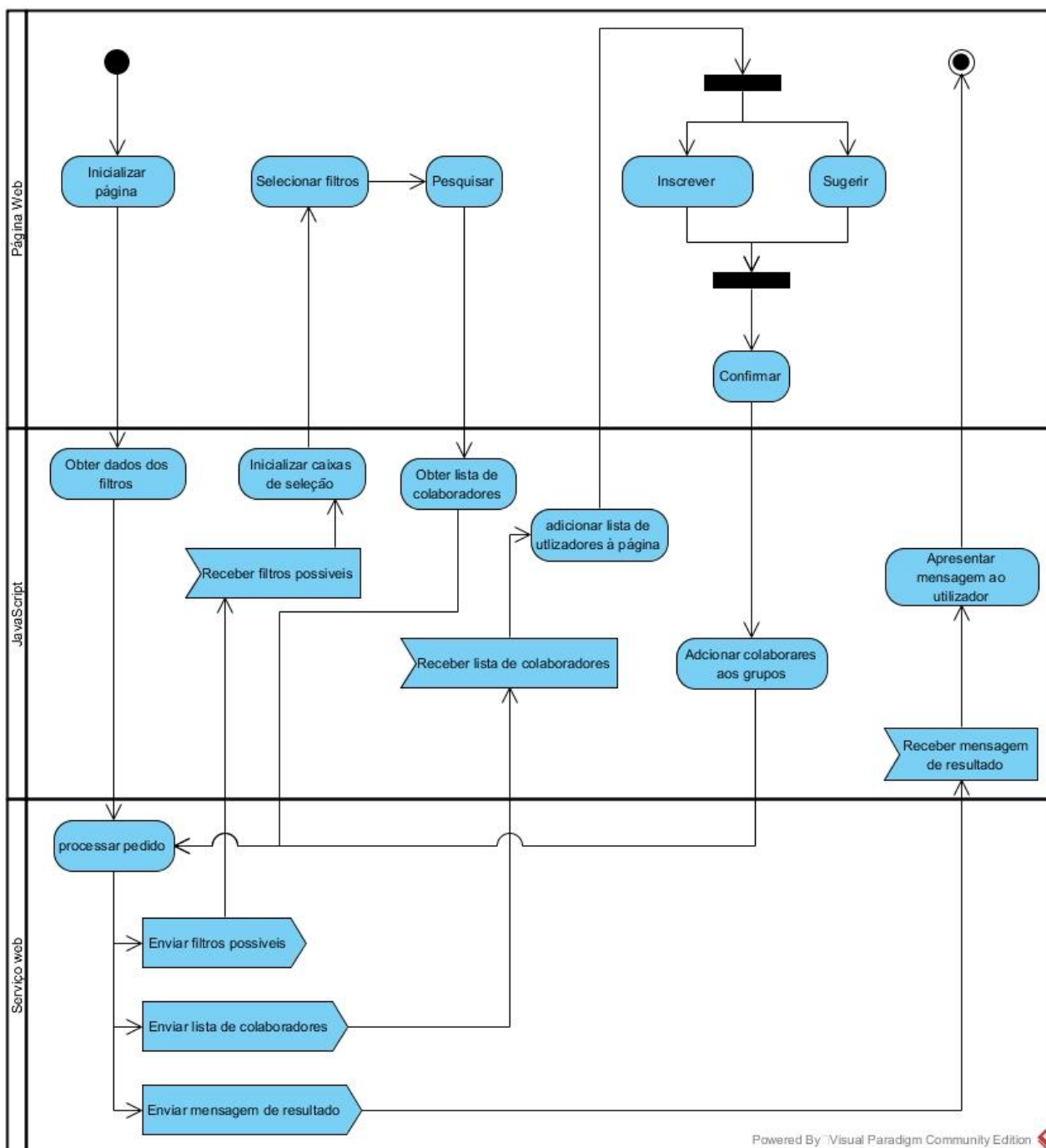


Figura 34 - Diagrama de atividades da página inscrição em cursos

### 4.3.2 Confirmar Sugestões

A página de confirmação de sugestões está presente em dois *sites* desta coleção: o *site* “Sugestões”, que se destina aos responsáveis de cada equipa e o *site* “Administração”, que se destina ao administrador funcional. Esta página tem como principal funcionalidade apresentar ao utilizador as sugestões de cursos para os utilizadores que não cumpriram os objetivos, para que este possa decidir se inscreve os utilizadores no curso ou se rejeita a sugestão do sistema.

Filtros

Cursos/Planos:

Escolha um ou mais

Seleccionar tudo...

Tag:

Escolha um ou mais

Seleccionar tudo...

Perfil:

Escolha um ou mais

Seleccionar tudo...

Estrutura:

Escolha um ou mais

Seleccionar tudo...

Área:

Escolha um ou mais

Seleccionar tudo...

Formando:

Escolha um ou mais

Seleccionar tudo...

Filtrar

Sugestões

Formando

Curso

Estrutura

Área

Tag

Média

Desvio

Insc. Limpar

Rej. Todas

Paulo Silva	Curso de exemplo v3	3060	Serviços...	Autoriza...	28.97	-37.25	<input checked="" type="checkbox"/>	<input type="checkbox"/>
José Fernando A Tomas	Curso de exemplo v3	4080	Agência...	Autoriza...	28.97	-78.43	<input checked="" type="checkbox"/>	<input type="checkbox"/>
João Paulo M Coutinho	Curso de exemplo v3	3310	Agência...	Autoriza...	28.97	-83.57	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Sofia Beirão	Curso de exemplo v3	3240	Agência...	Autoriza...	28.97	-75.35	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Joao Sousa	Curso de exemplo v3	1340	Agência...	Autoriza...	28.97	-13.7	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Sandra Rocha	Curso de exemplo v3	1460	Agência...	Autoriza...	28.97	-82.74	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Rute Marlene Teixeira...	Curso de exemplo v3	2160	Agência...	Autoriza...	28.97	-73.46	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Exportar

Confirmar

Figura 35 - Página de confirmação de sugestões

A página encontra-se dividida em duas secções, à semelhança das páginas apresentadas na secção 3.4.3 . A secção de filtros é composta por caixas de seleção onde é possível escolher um ou mais valores. A secção de sugestões apresenta uma lista de sugestões com um utilizador e a respetiva sugestão. De seguida são apresentadas informações do utilizador em especial o desvio que teve da média. No fim da linha, existem duas caixas seleção que permitem definir a aprovação ou rejeição da sugestão. Ainda na secção de sugestões é possível exportar para um ficheiro CSV (Comma Separated Values) e também executar a aprovação ou rejeição dos itens que estão visíveis na página.

A Figura 36 mostra o diagrama de atividades desta página. A página é inicializada com a lista de sugestões previamente carregadas e os valores disponíveis nas caixas de seleção da secção de filtros também são adicionados nesta fase. Quando a página está pronta o utilizador tem a possibilidade de realizar 3 ações: escolher os filtros e aplicar os mesmos; escolher aprovar ou rejeitar sugestões e aplicar; exportar todas as sugestões visíveis a página para um ficheiro CSV. Quando o utilizador excuta a confirmação de sugestões, o *script* transforma a lista que está visível num objeto JSON que é anexado ao pedido que é enviado ao serviço *web*.

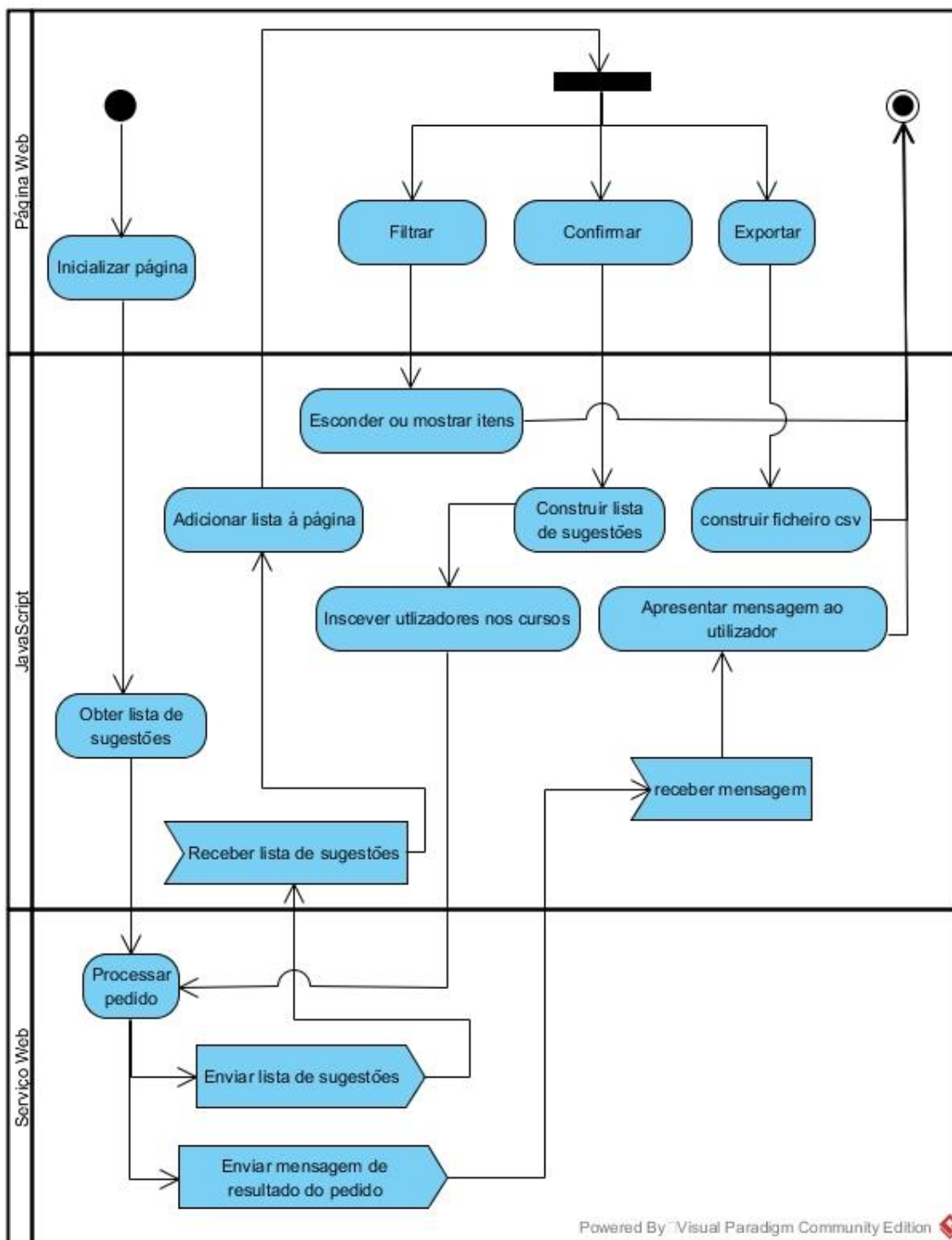


Figura 36 - Diagrama de atividades da página "Confirmar Sugestões"

### 4.3.3 Obter Sugestões

A página "Obter Sugestões", Figura 37, apenas disponível para o administrador funcional tem como principal objetivo agendar o processo que gera as sugestões. Adicionalmente esta também permite visualizar as informações sobre as configurações em que a ultima geração foi feita e em que a próxima será feita. Esta está dividida em duas secções. A primeira secção apresenta as informações sobre as configurações

utilizadas no processo de geração das sugestões. A segunda secção contém: uma caixa para definir o número máximo de sugestões por utilizador; uma caixa para definir os *e-mails* a serem notificados quando o processo terminar; um botão que irá agendar a execução do processo. Este processo é agendado automaticamente, mas o administrador funcional poderá, sempre que precisar, antecipar a execução deste processo, carregando no botão “Obter Sugestões Manualmente”.

## Obter Sugestões

Última Geração de Sugestões	Próxima Geração de Sugestões
Data de geração: 13/10/2017 12:20:40	Data de geração: 05/01/2018
Método de geração: Manual	Método de geração: Automático
Máx. sugestões : 5	Máx. sugestões : 5

---

### Configuração de Geração de Sugestões

Máximo Sugestões:	<input type="text" value="5"/>
Emails a notificar:	<input type="text" value="tmsantos@"/>
<input type="button" value="Obter Sugestões Manualmente"/>	

Figura 37 - Página para obter sugestões de cursos

### 4.3.4 Incumpridores de objetivos gerais

A página de incumpridores de objetivos, Figura 38, tem uma estrutura e funcionamento semelhante à página apresentada na secção 4.3.1 . A secção “Filtros” permite ao administrador funcional seleccionar o grupo de colaboradores que pretende obter, bem como um intervalo do desvio dos objetivos que estes obtiveram no último ciclo do sistema. A secção “Desvios por utilizador” apresenta uma listagem de utilizadores que é obtida através de um pedido ao serviço *web*. No fim desta secção encontra-se um botão que permite ao administrador exportar a listagem apresentada na página para um ficheiro CSV.



## GRO Geral

### Filtros

**Perfil:** Escolha um ou mais ▾  
 Seleccionar tudo...

**Estrutura:** Escolha um ou mais ▾  
 Seleccionar tudo...

**Área:** Escolha um ou mais ▾  
 Seleccionar tudo...

**Desvio:** Todos ▾

**Formando:** Escolha um ou mais ▾  
 Seleccionar tudo...

Filtrar

### Desvios por utilizador

Data a que se refere: 17/10/2017

Formando	Estrutura	Área	Perfil	Média	Desvio
Sónia Maria V G Afonso	5240	Porto de Mós	Comercial	28.08	-3.31

Exportar

Figura 38 - Página para listar incumpridores de objetivos

### 4.3.5 Configuração de sugestões

A página “Configuração de Sugestões”, permite que o administrador funcional altere as configurações do processo de geração de sugestões, sem iniciar o processo, como acontece na página “Obter Sugestões”. A primeira secção desta página, que inclui informações sobre a última geração de sugestões e próxima a ser executada e as configurações atuais do processo, é igual à página “Obter Sugestões”. A diferença entre as duas páginas é que nesta apenas se faz a alteração das configurações e não se agenda a execução do processo. Na secção seguinte são atribuídos intervalos de desvio da média que os colaboradores podem ter e qual é o peso do objetivo que devem cumprir. Estes valores são usados pelo sistema para priorizar as sugestões a serem feitas pois existe um limite de sugestões a serem feitas a cada colaborador. No fim de cada secção encontra-se um botão “Guardar” que, quando acionado, recolhe todos os valores presentes nas caixas de texto desta página e envia para o serviço *web* que atualiza na base de dados do módulo de integração as configurações de geração de sugestões.



## Configuração de Sugestões

### Última Geração de Sugestões

Data de geração: 13/10/2017 12:20:40  
Método de geração: Manual  
Máx. sugestões: 5

### Próxima Geração de Sugestões

Data de geração: 05/01/2018  
Método de geração: Automático  
Máx. sugestões: 5

### Configuração de Geração de Sugestões

Gerar sugestões x dias após o início do trimestre:

5

Aprovar Sugestões x dias após geração:

5

Máximo de Sugestões:

5

Emails a notificar:

lmsantos@

Guardar

### Ranking GRO

Rúbrica	Intervalo de tolerância %	Peso
Depósitos à Ordem	0	100
Depósitos a Prazo e Poupanças	81	100
Capital Social	0	8
Crédito Habitação	0	100
Crédito Pessoal	0	8

Guardar

Figura 39 - Página de configuração de Sugestões

## Capítulo 5 Conclusões

### 5.1 Trabalho desenvolvido

No Capítulo 3 foi abordada a ferramenta de reporte de horas, desenvolvida sobre o Sharepoint online. Abordou-se a arquitetura que foi construída para a implementação dos requisitos, com recurso a diagramas, e a metodologia usada para a implementação. Foram explicadas as interações pessoa-máquina, bem como as tecnologias usadas para construir essas interfaces.

No Capítulo 4 foi abordada a ferramenta de *e-learning*, construída com recurso ao Sharepoint *Online*, Sharepoint *On-premises*, LMS 365 e sistemas legados do cliente. Abordou-se, num nível mais abstrato, a arquitetura construída para a implementação dos requisitos e mais especificamente as interfaces pessoa-máquina que foram desenvolvidas.

### 5.2 Dificuldades

Realizar um estágio, num contexto empresarial e profissional, mostrou-se um importante desafio para finalizar este ciclo de formação. Revelou as dificuldades das constantes mudanças que existem neste contexto, que muitas vezes não são bem percebidas durante o percurso académico.

O maior obstáculo encontrado foi o Sharepoint, uma ferramenta Microsoft com o qual nunca tinha contactado. Esta ferramenta tem muitas funcionalidades e conceitos que têm de ser estudados para serem utilizados de forma eficiente. Aliado a esta dificuldade inicial, o tempo que foi dado para realizar os desenvolvimentos dificultou a aplicação de boas práticas. O resultado foi um maior número de erros no produto final que tiveram de ser resolvidos após a entrada em produção.

### 5.3 Conclusões

Com as evoluções tecnológicas e consequente evolução dos serviços prestados através da Internet pode observar-se uma grande tendência de que as empresas sejam empurradas para a utilização de serviços na nuvem, como por exemplo o armazenamento de dados. As vantagens de adotar estes modelos são evidentes, menos

preocupação com manutenção de infraestruturas, e isto leva a uma redução de custos de operação. As desvantagens, por outro lado, são preocupantes e muitas vezes aumentam a resistência em avançar para modelos com total utilização dos serviços na nuvem.

O Sharepoint é uma ferramenta muito utilizada por empresas porque oferece uma estrutura e funcionalidades para criar aplicações de colaboração e produtividade. Apesar de muitas das funcionalidades serem transversais às diferentes empresas, estas por vezes têm a necessidade de fazer alterações ou adicionar funcionalidades de forma a satisfazer exatamente as suas necessidades. O Sharepoint é disponibilizado numa versão online e numa versão *on-premises*. O Sharepoint Online oferece muitos serviços na nuvem que permitem às empresas construir as suas aplicações de colaboração sem terem infraestrutura para as manter. Por outro lado, o Sharepoint *on-premises* oferece mais opções e funcionalidades, mas as empresas tem de criar a sua infraestrutura ou contratar um serviço que a forneça. Durante a realização do estágio pude constatar que aliados a requisitos de segurança, a infraestrutura de rede da instituição, onde o trabalho foi realizado, precisa de ser melhorada para que os serviços na nuvem possam ser utilizados sem problemas.

Neste relatório é possível fazer a comparação entre dois projetos distintos, na mesma empresa. O projeto de reporte de horas, apesar de ser mais simples que o segundo, por não haver a necessidade de interagir com outros sistemas da empresa, refletiu as facilidades e vantagens apresentadas anteriormente para a criação de aplicações recorrendo a serviços na nuvem. O projeto de *e-learning* apresentou uma complexidade adicional, pois interage com outros sistemas para realizar as suas funções. Originalmente, a solução previa uma completa utilização de serviços na nuvem para construir o módulo de integração. Seriam criados processos que seriam executados na nuvem e que iriam realizar as ações de *backend*. No Sharepoint Online seriam criadas todas as páginas de interação com o utilizador. Este modelo foi parcialmente abandonado por não ser compatível com políticas de segurança e porque a infraestrutura de rede não estava pronta para receber uma utilização intensiva dos serviços na nuvem. Este projeto refletiu a intenção de usar este tipo de serviços e as suas vantagens, mas empresas que se apresentem as mesmas condições, que esta, podem não estar preparadas para adotar a nuvem na totalidade. Neste momento a Microsoft aconselha a construção de um ambiente híbrido para empresas que procuram diminuir os custos de operação, mas não conseguem fazer ou não podem fazer uma passagem completa dos

dados para a nuvem. Ou seja, utilizar o Office 365 apenas para estender as funcionalidades de infraestrutura já existente sem a descontinuar e, por exemplo, diminuir custos no armazenamento de dados dos colaboradores, sendo este armazenamento feito na nuvem.

## 5.4 Trabalho futuro

Os próximos passos para desenvolvimento de soluções à medida em Sharepoint *Online* passam pela utilização do Sharepoint Framework para desenvolver peças *web*.

O Sharepoint *Framework* oferece uma ferramenta genérica para criar interfaces customizadas bem como a construção de aplicações sobre Sharepoint Online. São utilizadas tecnologias *open source* (Node.js, Gulp, Webpack, Yeoman). Estas ferramentas são usadas para construir, empacotar e instalar as soluções. Uma grande vantagem de utilizar esta abordagem é que permite a programadores que nunca tenham interagido com tecnologias Microsoft facilmente produzam soluções para Sharepoint *Online*.



## Capítulo 6 Bibliografia

- [1] A. Perran, S. Perran, J. Mason, e L. Rogers, «Beginning SharePoint 2013 Building Business Solutions», p. 676, 2013.
- [2] «Overview of Business Connectivity Services in SharePoint 2013». [Em linha]. Disponível em: <https://technet.microsoft.com/en-us/library/ee661740.aspx>. [Acedido: 30-Jul-2017].
- [3] «Overview of records management in SharePoint Server 2013». [Em linha]. Disponível em: <https://technet.microsoft.com/en-us/library/cc261982.aspx>. [Acedido: 30-Jul-2017].
- [4] «Server and Site Architecture: Object Model Overview». [Em linha]. Disponível em: [https://msdn.microsoft.com/en-us/library/office/ms473633\(v=office.14\).aspx](https://msdn.microsoft.com/en-us/library/office/ms473633(v=office.14).aspx). [Acedido: 12-Set-2017].
- [5] «A Short History of JavaScript - Web Education Community Group». [Em linha]. Disponível em: [https://www.w3.org/community/webed/wiki/A\\_Short\\_History\\_of\\_JavaScript](https://www.w3.org/community/webed/wiki/A_Short_History_of_JavaScript). [Acedido: 22-Ago-2017].
- [6] Unicode Consortium., *The Unicode standard*. Addison-Wesley, 2000.
- [7] «Standard ECMA-262». [Em linha]. Disponível em: <https://www.ecma-international.org/publications/standards/Ecma-262.htm>. [Acedido: 22-Ago-2017].
- [8] «New in JavaScript - JavaScript | MDN». [Em linha]. Disponível em: [https://developer.mozilla.org/en-US/docs/Web/JavaScript/New\\_in\\_JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript/New_in_JavaScript). [Acedido: 22-Ago-2017].
- [9] «JavaScript | MDN». [Em linha]. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. [Acedido: 23-Ago-2017].
- [10] «Prototype-based programming - Glossary | MDN». [Em linha]. Disponível em: [https://developer.mozilla.org/en-US/docs/Glossary/Prototype-based\\_programming](https://developer.mozilla.org/en-US/docs/Glossary/Prototype-based_programming). [Acedido: 23-Ago-2017].
- [11] «First-class Function - Glossary | MDN». [Em linha]. Disponível em: [https://developer.mozilla.org/en-US/docs/Glossary/First-class\\_Function](https://developer.mozilla.org/en-US/docs/Glossary/First-class_Function). [Acedido: 23-Ago-2017].
- [12] «jQuery». [Em linha]. Disponível em: <https://jquery.com/>. [Acedido: 24-Ago-2017].
- [13] «How jQuery Works | jQuery Learning Center». [Em linha]. Disponível em: <https://learn.jquery.com/about-jquery/how-jquery-works/>. [Acedido: 24-Ago-2017].
- [14] «jQuery() | jQuery API Documentation». [Em linha]. Disponível em: <http://api.jquery.com/jQuery/#jQuery-selector-context>. [Acedido: 24-Ago-2017].
- [15] «.on() | jQuery API Documentation». [Em linha]. Disponível em:

- <http://api.jquery.com/on/>. [Acedido: 24-Ago-2017].
- [16] «jQuery.ajax() | jQuery API Documentation». [Em linha]. Disponível em: <http://api.jquery.com/jQuery.ajax/>. [Acedido: 25-Ago-2017].
- [17] «List and library column types and options - SharePoint». [Em linha]. Disponível em: <https://support.office.com/en-us/article/List-and-library-column-types-and-options-0d8ddb7b-7dc7-414d-a283-ee9dca891df7>. [Acedido: 15-Set-2017].